Robotica (2008) volume 26, pp. 41–53. © 2007 Cambridge University Press
doi:10.1017/S0263574707003566 Printed in the United Kingdom

Optimal arrest and guidance of a moving prismatic object using multiagents

Pankaj Sharma†, Anupam Saxena‡ and Ashish Dutta§*

(Received in Final Form: May 8, 2007. First published online: October 12, 2007)

SUMMARY

Genetic algorithm is used to determine the optimal capture points for the multi agents required to grasp a moving generic prismatic object by arresting it in form closure. Thereafter, the agents approach their respective moving goals using a decentralized projective path planning algorithm. Post arrest, the object is guided along a desired linear path to a desired goal point. Form closure of the object is obtained using the concept of accessibility angle. A convex envelop is formed around the object, and the goal points on the object boundary are mapped onto the envelope. The robots approach the mapped goal points first, and then, converge on the actual object. This ensures that the agents reach the actual goal points almost simultaneously, and do not undergo looping at a local concave region. The object is assumed alive while being captured but is assumed compromised thereafter. Post arrest, robots alter their positions optimally around the object to transport it along a desired direction. Frictionless point contact between the object and a robot is assumed. The shape of the mobile robot is considered cylindrical such that it can only apply force along the outward radial direction. Simulation results are presented that illustrate the effectiveness of the proposed method.

KEYWORDS: accessibility angle; multi agents; projective path planning; genetic algorithm

1. Introduction

The study of multiagent capture and manipulation of an object has been an area of active interest for many researchers. Although the notion of swarm robotics is borrowed from nature, such as behavior in ant colonies and bee hives, etc, it has many applications in robotics. Some of them include automated highway systems, formation flight control, unmanned underwater vehicles, satellite clustering, exploration, surveillance, search and rescue, mapping of unknown/partially unknown environments, distributed manipulation, and transportation of large objects. This paper addresses a novel approach for catching a moving object with the help of multiagents in an optimized grasp configuration using the proposed optimization function. Agents are assumed to apply force only along the direction normal to their cylindrical envelop. Once the optimized grasp points, and thus, the number of robots are determined, robots are sent, from their initially arbitrary positions to those optimal grasp points on the boundary of the object. Robots employ the proposed *projective path planning* algorithm to reach their goal points by avoiding collisions with the environment. After the object is arrested, it is considered lifeless. Robots again rearrange themselves around the object in an optimized configuration so that they can apply force along the desired direction. Object is then transported to the desired location by multiagents in broken serpentine motion.

The main contributions of this paper are: (a) optimal grasp points, in form closure configuration, for capturing a moving object are found that gives the minimum force required to constrain the object using mobile robots, (b) the minimum number of robots are found that are required for constraining and then optimally transporting the object and (c) a novel potential field free and deterministic projective path planning method is developed that always guarantees collision free paths for the mobile robots. The proposed problem in swarm robotics borrows several ideas from the well-known problem in robotics grasping. However, there are several differences between finger motion and mobile robot motion, such as mobile robots are autonomous and can move in any direction without colliding with the object or other robots. While fingers are constrained by their kinematics and are not free to move in all directions. Another difference is that fingers cannot freely switch their functions and reorient like mobile robots for pushing an object in a desired direction, again due to the kinematic constraints.

A detailed review of multi-finger grasping and manipulation can be found in Bicchi and Kumar.[1] Yoshikawa[2] proposed conditions for passive and active closure by a constraining mechanism. Rimon and Blake[3] suggested methods to capture 2D objects by a two-fingered gripper so that the object cannot escape the cage. Davidson and Blake[4] extended this concept to capture two-dimensional (2D) objects with one parameter three-finger grippers. Ding *et al.*[5] proposed a method for computing form closure grasp on polyhedral objects. Ponce and Faverjon.[6] worked on calculating three-finger force closure grasp for flat objects. Kaneko *et al.*[7] proposed an approach for stopping an object under dynamic friction closure due to coulomb's friction. Aiyama *et al.*[8] showed cooperative transport of an object by two four-legged robots. Each robot uses its own sensor only to know the position of the object and intentional direction of movement of the other robot. Hashimoto *et al.*[9] proposed

† Department of Mechanical Engineering, Indian Institute of Technology, Kanpur 208016 INDIA.
‡ Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850, USA. E-mail: as574@cornell.edu
§ Department of Mechanical Engineering, Nagoya University, Chikusa-ku, Furo-cho, Nagoya 464-8603, Japan.
* Corresponding author: E-mail: adutta@iitk.ac.in

an algorithm for coordinated transfer of a heavy object by several wheeled mobile robots. Hashimoto and Uminoroib[10] proposed a dynamic control approach for the motion coordination of multiple wheeled robots. Ahmadabadi and Nakano[11] proposed an approach for a group of mobile robots to manipulate an object at rest. The task is divided into two subtasks, "constrain" and "move." Based on this idea, the object can be carried along a straight line or/and can be moved about a fixed point. Yamada and Saito[12] proposed a method for pushing a box by multiple robots without explicit communication for dynamic environment. Sun and Mills[13] developed a method for payload manipulation using a compliant gripper. It is shown that a PD position feedback plus gravity compensation controller can regulate the desired position and orientation of the payload. Galta *et al.*[14] showed that for a stable grasp with three contact points, the point of intersection of the contact normals should be close to the geometric center of the object, and the grip span should be maximum.

Belta and Kumar[15] proposed an algorithm for controlling a large number of robots moving in a group. This allows the number of robots to be very large in a mission. Burgard *et al.*[16] suggested the problem of exploring an unknown environment by a team of robots such that the total time of exploration is minimized. Sugar and Kumar[17] proposed an algorithm for controlling mobile manipulators focusing on tasks that require grasping, manipulation, and transporting large and flexible objects. The robots can cooperatively transport the object and march in a tightly controlled formation while having the ability to navigate autonomously. Spletzer *et al.*[18] proposed a framework to maintain the relative positions and orientations of swarms during cooperative tasks. Carpin and Parker[19] introduced an algorithm for cooperative multirobot team with heterogeneous sensing capabilities allowing the robots to remain in formation while interacting with other obstacles within the formation. Fierro *et al.*[20] described a framework for controlling a group of nonholonomic mobile robots equipped with a range sensor. The vehicles are required to follow a prescribed trajectory while maintaining the desired formation. Fierro and Das[21] proposed a model for reconfigurable robots using hybrid control. The robot team can reconfigure themselves dynamically while following a prescribed trajectory. Sudsang and Ponce[22] proposed a framework in which three disc-shaped robots manipulate a polygonal object on a plane in the presence of obstacles. The disc-shaped robots form closure such that the object cannot escape. Wang and Kumar[23] devised a way to transport a polygonal object-by-object closure. They used first-order potential field-based controller for this method. Buttazzo *et al.*[24] proposed and implemented a method by which a robot can trap a moving object using visual sensor information. They could catch an object moving with speed up to 700 mm/s and 1500 mm/s^2 acceleration. Liu *et al.*[25] proposed a method for dynamically intercepting and manipulating the object for industrial robots in a work cell. The robot uses information from vision camera. Song and Kumar[26] proposed a potential field-based approach for multi-robot manipulation.

The main contributions of our work with reference to the research mentioned earlier are on the three issues of (a) nature of the system (static or dynamic object capture), (b) utilization of resources (number of mobile robots used), and (c) distributability (centralized or decentralized control).

Majority of earlier researchers have considered a static object that is pushed[3,4,23,26] or lifted and transferred[8,10,11,17,24] to a desired goal point. In the first case, the object is surrounded by a large number of mobile robots, and then, pushed to the desired place. In the second case, the robots lift the object by constraining it between robots or using grippers. In our proposed method, the object is not stationary but moves in a predefined path, and the mobile robots first, optimally constrain the object, and then, move it to the desired goal. To constrain the object in form closure, the concept of accessibility angle is used. This enables us to get fail proof grasps that can be found quickly. The optimal positions where the robots should go and hold the object are also found. Hence, this method has several new advantageous features like capturing a gradually moving object. The optimal grasp points are found such that the forces required for holding the object will be the least.

Utilization of resources is an important parameter that most past researchers have not considered. This is so because using a large number of robots to constrain and move an object leads to increased energy consumption, traffic routing problems, communication with host controller, etc. Several past researcher have used a very large number of robots to surround and move an object[10,19,20,21,23] while others have used lesser robots (about four) to lift and transfer an object.[8,11,17] A major contribution of this paper is that the optimal object contact points are determined, and correspondingly, the number of robots required to constrain the object and push it to a desired location is also optimized. The pattern in which the robots switch positions to take the object to the desired location is also optimized. Hence, we require the least number of robots to complete the constraining and transporting of the object.

Most of the earlier researchers dealing with a large number of robots used the potential field approach to plan the path.[10,20,21,23,26] In case of collisions, the robots move away from each other. In our case, we have used a centralized global planner that first checks for the most optimal goal point for each robot, and then, checks for and avoids collisions using projective path planning. This is deterministic in nature and always ensures a collision free path. The object is assumed to move in a predefined path, and hence, its position is always known. Otherwise, if the object moves to escape the multiagent grasp, its motion pattern can be captured at each time interval. Based on the current and subsequent positions of the object for a time interval, the local collision-free positions of the agents for that time interval is planned.

In this paper, the procedure adopted for optimal placement of robots around the object is explained in Section 2. The objective function and the constraints used are also explained in this section. Various aspects of the projective path planning algorithm used to move the robots towards their moving goal points are explained in Section 3. Optimal repositioning of the robots after the moving object is compromised to guide it along a straight line towards its destination is described in Section 4. Results are presented in Section 5, and some concluding remarks are made in Section 6.

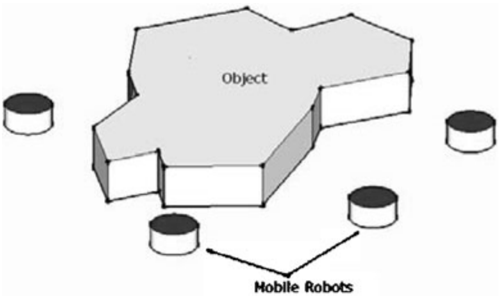

Fig. 1. The object contour and the mobile robot geometry; object is prismatic and multiagents are cylindrical in shape.

2. Optimized Capture of a Slowly Moving Object

The first problem is to find the best possible goal points on the object boundary where the robots can be sent for an efficient grasp, after which the object is neither able to translate nor rotate freely. The general layout of the problem is shown in Fig. 1 that shows a prismatic object and four mobile robots. There are many possible ways of having form closure of an object in which the number of contact points, and thus, the number of constraining mobile robots may vary. Hence, optimization is used to determine the minimum number of robots and the best goal points on the object that satisfy certain constraints. By optimized catch of the object is meant the configuration of multiagents around the object boundary such that the normal forces required for grasping the object against any external forces or moments are minimal, with the constraint that the object is in *form closure*. The basic definitions of form and force closures used are as given by Yoshikawa.[2] The object to be caught and taken to a desired location is modeled as a prismatic object, that is, with a 2D polygonal contour extruded in the third dimension. In this work, it is assumed that the surfaces of the object as well as the robots are not deformable implying frictionless line contact between the two surfaces. This assumption models the force interaction between the object and the robots such that the force is always normal to the object boundary.

2.1. *Accessibility angle constraint*

Freedom angle can be defined for a point on a 2D object boundary where the contact exists. The freedom angle is a possible range of directions from that particular point along which the object can be taken away without any intervention with that point if it is considered as stationary and rigid on its place. It can be calculated if the contact point's location and the object shape are known (Fig. 2). Accessibility angle[27] depends on the freedom angle of an object at a point. Freedom angle can further be classified as *global* or *local* freedom angles. For an object with polygonal contour, the span of local freedom angles (i.e., difference in maximum and minimum ends of the range of angle) would always be 180° when the contact point is on the line and not on one of the vertices.

Accessibility angle for an object for some grasp configuration is the common angle (intersection set) between the freedom angles at all contact points. For computing the accessibility angle, first, freedom angles (ranges) at all contact points are determined. Intersection of all freedom angles (ranges) is known as the accessibility angle (range) for the object. If any such range exists, the object is accessible in translation, or in other words, can be displaced within that angular zone.

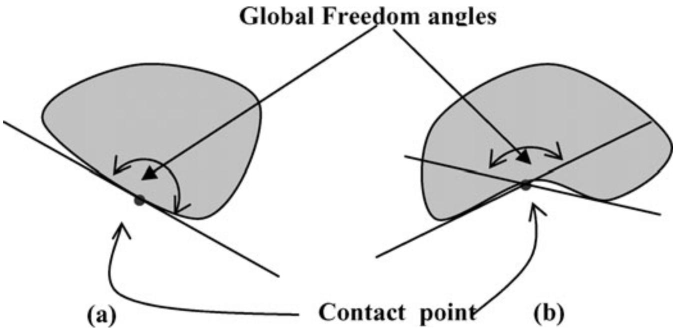

Fig. 2. (a) Global freedom angle for a convex shaped object. (b) Global freedom angle for a concave shaped object.

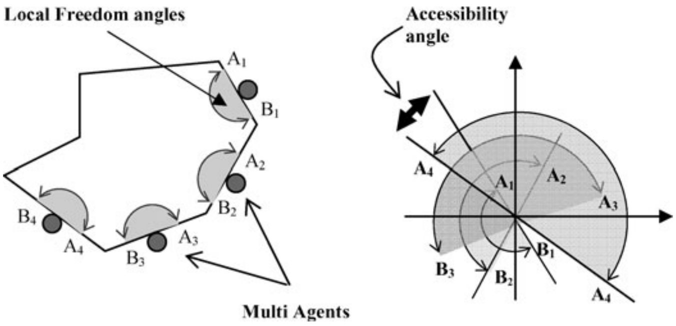

Fig. 3. (a) An object is surrounded by four mobile robots, the local freedom angles at the four contact points are shown in dark semicircles. (b) Accessibility angle is shown. There exists an angular region which is the intersection of all freedom angles; the object can escape from within that region since it is not fully constrained in translation.

Figure 3 shows an example of a polygonal object whose local freedom angles are shown using dark semicircles. Note that the object is not totally arrested in translation, and there is an angular region (range shown with dark arrows) such that the object can be pulled out along any direction within this region. Thus, for an object to be fully constrained in translation, there should not exist any common angle between all the freedoms angles, i.e., the intersection set between all freedom angles should be null. Local accessibility for an object is calculated based on the local freedom angles at the points whereas global accessibility is calculated based on the global freedom angles. Global accessibility angle accounts for an object's escape in translation from the catch as a whole, while it may allow the object to be displaced somewhat even in catch. If we consider the catch using local accessibility angle based on the local freedom angles, this ensures that the object will not be able to move even slightly when in catch. An example of such a situation is shown in Fig. 4.

In this work, we use local accessibility angles for polygonal shaped objects to ensure that the object cannot move even slightly when in catch. In general, if $\mathbf{r}_i$ is the position vector at the ith point on the object boundary (see Fig. 5) and is a candidate grasp point, the local freedom angle at that point is given by Eq. (1) wherein $\angle\mathbf{a}$ represents the angle of vector $\mathbf{a}$ from the horizontal, and the freedom angle is ϕ_{i}. Here, $\mathbf{r}_{\mathrm{i}+1}$ and $\mathbf{r}_{\mathrm{i}-1}$ are position vectors of vertices such that $\mathbf{r}_{\mathrm{i}}\mathbf{r}_{\mathrm{i}-1}$

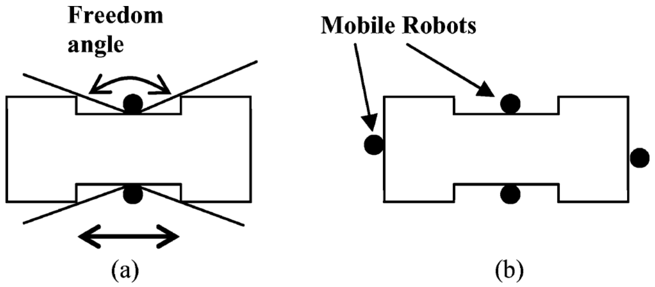

Fig. 4. (a) Figure shows that even two contact points are sufficient for constraining the translation accessibility but object will translate slightly along the horizontal direction. (b) Uses local accessibility concept which requires four contact points for the same object to constrain all possible translations and it does not allow object to move even slightly.

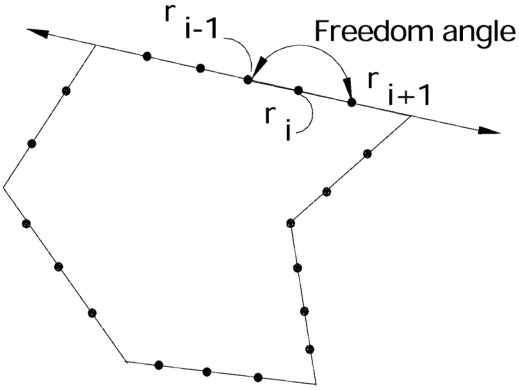

Fig. 5. The points on the object boundary are shown which can be the possible candidates for the goal points for mobile robots.

and $\mathbf{r}_i\mathbf{r}_{i+1}$ are tangents to the object from the candidate grasp point. To compute the local freedom angle, $\mathbf{r}_{i+1}$, $\mathbf{r}_i$, and $\mathbf{r}_{i-1}$ are usually consecutive.

$$\phi_i = \{< (\mathbf{r}_{i+1} - \mathbf{r}_i), < (\mathbf{r}_{i-1} - \mathbf{r}_i)\} \tag{1}$$

$$\psi = \{\phi_1 \cap \phi_2 \cap \ldots\ldots \cap \phi_n\} \tag{2}$$

Accessibility angle is given as Ψ, as in Eq. (2), which is calculated by finding the intersection between freedom angles at all the catch points for an object. Here, n is the total number of catch points (and the number of agents) on the object. The accessibility angle constraint helps in eliminating the grasp configurations for which form closure of the object is not attained.

2.2. *Other constraints*

Several other constraints are enforced to obtain the desired positions of the robots on the object. To have moment closure, swarms should be placed such that both clockwise and anticlockwise external moments can be resisted. Thus, the object cannot rotate in any direction. Reaction moments caused by the normal reaction forces applied by the mobile robots is calculated about the center of gravity of the object. However, this constraint fails on circular-shaped objects or on regular polygons because all normals at the boundary are coincident. Therefore, such objects cannot be constrained in rotation in case of frictionless contact; they can only be constrained in translation. Other constraints are modeled such that robots do not get clustered in one region as well as their positions do not overlap with each other. The robot is assigned a hypothetical envelope, which takes care of robots not intersecting at the object boundary. To allow better distribution of swarm robots around the boundary, no two robots are allowed to catch the object on the same edge. A constraint also ensures that the mobile robots as a group are capable of resisting the moments in both clockwise as well as anticlockwise direction. Any configuration wherein robots cannot resist moment in any direction is rejected.

2.3. *Optimization function*

The objective is to obtain an optimal grasp configuration such that (a) resisting moments from the agents in both directions is maximized, (b) the number of agents is minimized, (c) the agents contribute equally to resist clockwise and counterclockwise moments, and (d) all the above constraints, and most importantly, the form closure constraint are met. It is assumed that an agent can only impart a limited resisting force on the object. In cases where an object imposes high magnitudes of moments to be resisted by the agents, e.g., when a moving object takes a sharp turn, it should be preferred that the agents get placed as far away as possible from each other. If unit forces are considered to be the maximum forces that can be applied by the agents along the boundary normals of the object in order to maximize resisting moments, their moment arms should be maximized. Consider M_{cw} as the sum of moments about the center of gravity of the object caused by robots in clockwise direction and M_{ccw} as that in the counter clockwise direction. Let N be the total number of mobile robots in the task. Also, let N_{cw} be the number of agents countering clockwise moment and N_{ccw} be those opposing counter clockwise moment about the center of gravity such that $N = N_{cw} + N_{ccw}$. To fulfill objective (a), the sum of absolute values of the moments, i.e., $|M_{cw}| + |M_{ccw}|$ can be maximized. This will, in turn, minimize the normal contact forces between the agents and the object. For (b), N^k can be minimized where k is the user specified parameter that controls the relative weight of this objective. To attain goal (c), $|N_{cw} + N_{ccw}|$ can be minimized to encourage a grasp configuration where the numbers of robots resisting clockwise and anticlockwise moments are the same. The three multicriteria objectives can be combined into a single one, which can be posed as

$$\text{Maximize} f = \left(\frac{|M_{cw}| + |M_{ccw}|}{N^k}\right)\left(\frac{1}{|N_{cw} - N_{ccw}| + \varepsilon}\right) \tag{3}$$

where small positive term ε in the denominator ensures that the denominator in the second term does not approach zero.

Note that the objective in Eq. (3) can be decoupled into (i) maximizing the sum of clockwise and anticlockwise moments and (ii) minimizing the number of agents and a multiobjective algorithm can be used to obtain a set of pareto optimal solutions. Use of multiobjective optimization will be investigated in future though to avoid choosing a solution from among the pareto optimal ones, which can be subjective and may require user interpretation; in this work, a genetic algorithm is used.

2.4. *Object boundary segmentation*

A mobile robot can approach the object boundary at any position. To model some finite points which are the possible candidates for goal points for mobile robots for the purpose of capturing the object, the boundary of the object is segmented into a set of finite nodes. An added advantage is that the optimization problem is modeled with these finite approach positions as discrete variables. This reduces the search space considerably, and allows using efficient stochastic algorithms for optimization. The placement of finite nodes on the boundary can be varied with the size of the object.

2.5. *Genetic algorithm*

Genetic algorithm[28] is used for optimizing the objective in Eq. (3) with the constraints mentioned in Sections 2.1 and 2.2. Genetic algorithm is used because it suits the discrete nature of the design variables formulated earlier. Note that the gradient-based algorithms cannot work with discontinuous variables. Among the constraints, if any is violated, a very large negative penalty of -10^{20} is imposed on the objective. This gradually eliminates the nonfeasible solutions in the population of candidate solutions, and tends to move towards feasible space for maximizing the function value.

The design variable vector used in genetic algorithm is a binary string. The length of the string is equal to the number of discretized points on the boundary. Each point is a possible goal point for a robot that it can approach for constraining/transferring the object. A value of '0' in the string represents no robot at that point and '1' represents presence of robot at that point. Details of simulation results for finding the least number of capture points are presented by Sharma *et al.*[29] The results of MATLAB simulation for optimized grasp for an arbitrary polygonal object are shown need. The parameter setting for optimizing the function is as follows: population size 60, mutation probability 0.12, crossover probability 0.8, and $k = 1$. When a big/heavy object is to be handled, the number of team members can be increased by varying the parameter k in the objective function. As k decreases, the number of catch points increases. We demonstrate optimality through Figs. 6–9 that depict the behavior of the two subobjectives in Eq. (3)

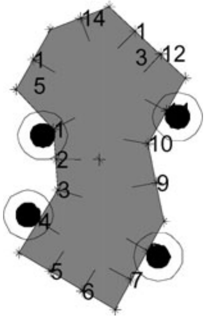

Fig. 6. The optimal solution. Figure shows a candidate prismatic object with four optimal grasp points (number of agents) to be determined. Time taken for optimization with MATLAB^{TM} in Intel P4 (2.2 GHz) is 156 s.

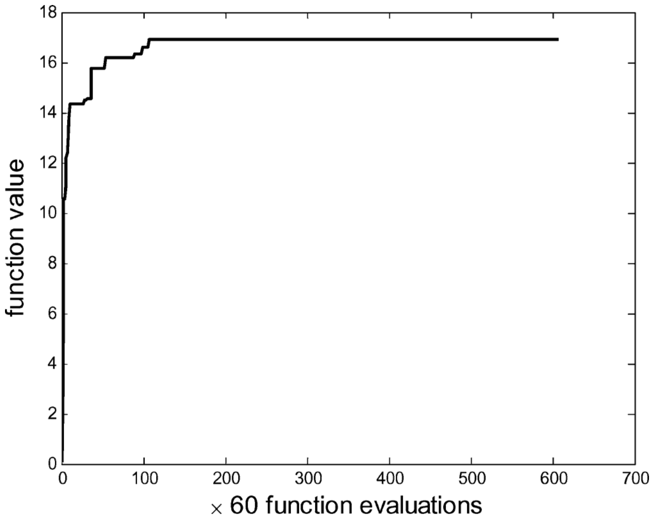

Fig. 7. Variation of the overall objective which gets maximized to 16.92 in 36000 function evaluations.

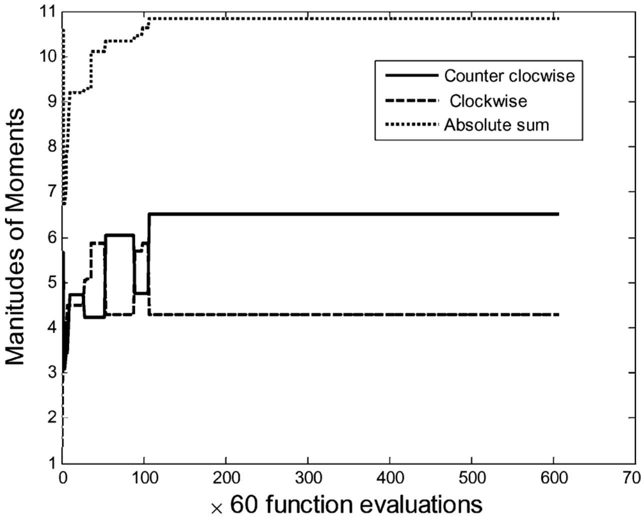

Fig 8. Variation in magnitudes of the overall, counterclockwise and clockwise moments with function evaluations.

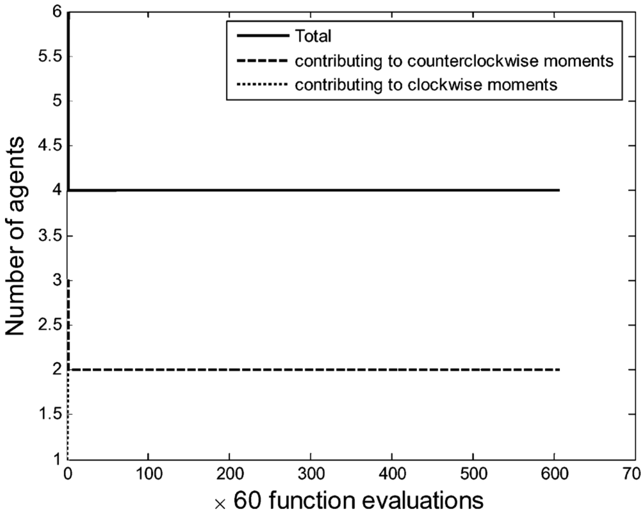

Fig. 9. Variation in the number of agents used for grasping with function evaluations. Graphs for individual number of agents contributing to counterclockwise and clockwise moments overlap.

with the number of function evaluations. Figure 6 shows a candidate prismatic object for which the four goal points are determined using the genetic algorithm. The filled circles show the optimal location of robots on the object boundary. Note that for an object in form closure, it has to be grasped at four points on its boundary. The overall objective in Eq. (3) is maximized to 16.92 requiring 36000 function evaluations in 156 s with MATLAB^{TM} executed using an Intel P4 2.2 GHz processor (Fig. 7). The infeasible solutions (for which the function is penalized) are not shown in the plots. Figure 8 depicts how the magnitudes of the counterclockwise and clockwise moments individually get maximized over the functions evaluated and Fig. 9 shows decrease in the total number of grasp points, and hence, agents to be used in form closure. Decision on the number of agents is reached much earlier in optimization after feasible solutions commence to emerge. Agents contributing to counterclockwise moments decrease from three to two while those contributing to clockwise moment first increase from one to three, and then, decrease to two. Their total number converges to four.

3. Projective Path Planning

Once the optimal grasp configuration for a given object and the number of agents are obtained, the novel projective path planning algorithm[30] is implemented as a global planner to plan the paths of multiagents to capture the moving object. The initial positions and orientations of the multiagents are known/arbitrary. The main feature of this algorithm is that it is deterministic in nature while interacting with the moving object and other moving robots so that the solution provided is always collision free. Most of the times, this algorithm allows multiagents to steer and move along the path with certain turning radius. This is helpful in avoiding sharp turns while catching the object. However, there exist situations when it is a must for a robot to take sharp turns to avoid collision with the moving object or other mobile agents.

Robots are assigned respective goal points to completely constrain the object satisfying form closure. Initially, a convex envelope around the object is formed. Size of this envelope is slightly larger than the object. The actual goal points (on the object) are mapped to the convex envelope (e.g., Fig. 10). When the robots reach the mapped goal points on the envelope, they are directed to the actual configuration points on the object boundary. Through this envelop, it is ensured that object is first surrounded by the robots before making the form closure grasp. This is also helpful in avoiding incessant looping of a robot that can be instigated around a concave vertex, thereby, hampering it from approaching its goal position.

The object to be captured is assumed to be moving on a plane. The path of the object can be along any generic curve. The projective path planning algorithm plans the path of multiagents based on the current and the successive position of the object. In each time step, the object moves to some other position with some finite speed. Subsequent robot positions are computed depending on the object's new position and their environment. The object may also perform escaping maneuvers to avoid the catch by multiagents. Initial position and orientation of a multiagent can be obtained through user input or can be arbitrary. Each robot is assigned its own goal point on the boundary of the object to access.

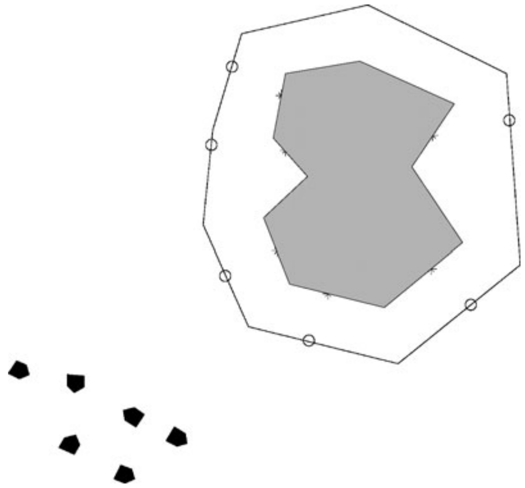

Fig. 10. Figure shows the convex envelope around the object. The goal points on the object boundary are marked with '*' while the mapped goal points are marked with 'o' on the convex envelop surrounding the object. When all agents reach the mapped goal points, they further move towards the actual goal points on the boundary.

3.1. *Impeding the object*

When a robot reaches its respective goal point on the envelope, it aligns its motion with the object's motion maintaining close distance to it until all other robots reach their respective goal points on the envelope. Thereafter, they start moving toward the actual goal points on the object boundary. This avoids incessant looping of robots around the object, and also, robots reach the object almost at the same time. As soon as a robot reaches near its goal point, it is recorded. If all other robots have also reached their respective goal points on object within certain tolerance, then the object is considered compromised.

3.2. *Modeling robot motion toward the goal point*

A robot's path is determined by a set of rules depending on the obstacles found in its path. At each step, a robot's movement towards its respective goal point (on the envelope or the object boundary) can be divided in two steps: (i) to translate the robot along its current orientation by a certain amount and (ii) to turn the robot in appropriate direction by an appropriate angle. Figure 11 shows a candidate path of the robot guided to its goal point. The amount of translation along its current orientation depends on the speed of the robot. The direction and amount of rotation are calculated based on the position of the robot relative to its respective goal point and other peer robots.

At any stage, let the position of the robot be represented by the position vector $\mathbf{O}_i$ and the orientation be represented by vector $\mathbf{D}_i$, which is a unit vector directing outward of robot along its tip direction. Let $\mathbf{O}_G$ be the position vector of the respective goal point on the envelope or on the object boundary.

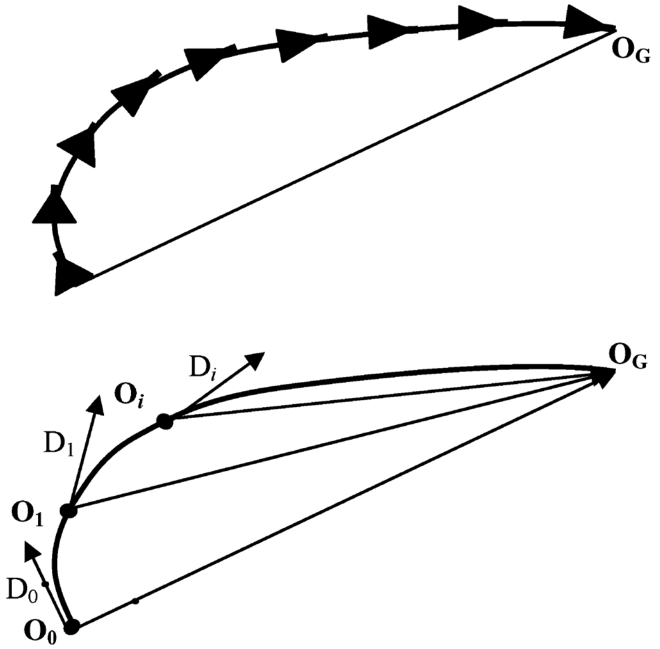

Fig. 11. A mobile robot moves towards the goal point. At each step, it calculates the sign of the cross product between the vector D$_i$ and O$_i$ O$_G$. D$_i$ is the current orientation of the agent, O$_i$ is its current position, and O$_G$ is the goal point the agent seeks.

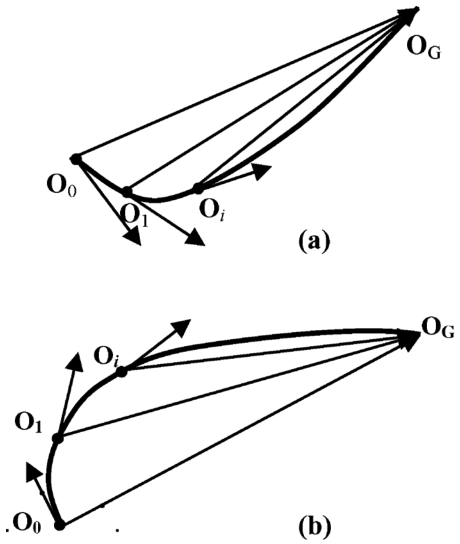

Fig. 12. (a) Robot takes left turn while trying to reach the object for positive sign of the cross product. (b) Robot takes right turn in case of negative cross product.

Let
$$\theta_i = \sin^{-1}\left(\frac{\mathbf{D}_i \times (\mathbf{O}_G - \mathbf{O}_i)}{|\mathbf{D}_i||(\mathbf{O}_G - \mathbf{O}_i)|}\right) \qquad (4)$$

where ‘×’ represents the cross product of vectors. If the sign of the cross product is positive, the robot is commanded to take the left turn from its current orientation and if it is negative, then the robot takes the right turn (see Fig. 12).

Rotation in each step is the minimum of two angles: (i) angle θ_i between the two vectors $\mathbf{D}_i$ and $\mathbf{O}_i\mathbf{O}_G$ and (ii) maximum allowable turn angle for a step. The maximum allowable turn angle depends on how sharp a turn a robot can take assuming it to be non holonomic. In this work, it is taken as 15°. When a robot aligns itself towards the goal point after some steps, it approaches the goal point along a straight line provided there is no obstacle in that path.

3.3. Projective collision avoidance modeling

To model collision between robots, each robot is considered to have a circular envelop of radius *r.* If the distance between centers of the two robots is C_1C_2, where C_1 and C_2 are centers of the two robots, then, if $C_1C_2 < 2r$, the two robots will collide with each other. Collision between the object and robot is modeled by considering the intersection between a circle and a set of lines (edges of the object's contour). The point(s) of intersection can be found by using basic geometric relations. When any edge of the object polygon is found intersecting with the envelope circle of the robot, collision is reported.

For a robot, to avoid collision with the object and also to trace its goal point simultaneously, certain rules are formulated as given in Table I. In a newly proposed method, the circular envelop of the robot is projected to the next step, and then, possibility of intersection with the polygonal contour of the object's current position is reviewed. Decisions are taken to change the orientation of the robot by updating its orientation angle. At each instant, the robot projects itself to the next position with a specified distance, the projection distance. The projection distance of a robot is decided relative to the radius *r* of its envelope, and is kept in the range of *2r* to *3r*. The reason for choosing projection distance in this range is because if a larger projection distance is chosen and if a robot collides somewhere between the current position and the projected position, it will not get noticed. If a robot finds no obstacle in its path at the subsequent step, it moves to the next step as per the rules framed for reaching a goal point. If it finds the object, the following action is taken.

The robot in its projected position is reoriented to align with the edge with which it is intersecting. Again, there can be two possibilities when aligning with the intersecting edge. Out of the two possible directions $\mathbf{E}_1$ and $\mathbf{E}_2$ as shown in Fig. 13, the one which has positive value of the dot product with the current orientation of robot ($\mathbf{D}_i$) is chosen. This takes care of the robot taking less sharp turn to avoid colliding with the object.

In case of intersection between the robot's projection and object polygon in current position so that the robot envelope intersects with multiple edges as shown in Fig. 14, the current orientation of the robot is aligned along the angular

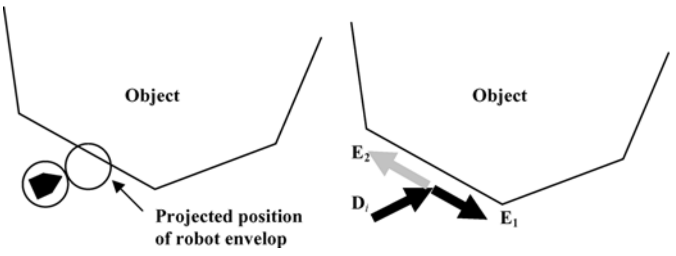

Fig. 13. The possible directions of robot orientation to avoid collision with the object ($\mathbf{E}_1$ and $\mathbf{E}_2$). Finally the direction shown in dark is chosen ($\mathbf{E}_1$) as the dot product of robot's current direction $\mathbf{D}_i$ with that direction being positive.

Table I. Decision rules for avoiding collision of an agent with the moving object and with other agents.

	Action taken		
Situation an agent faces	Move along the direction $\mathbf{D}_j$ of the respective goal point		
No obstacles in the path (object or other agents) Collision with the object	With any one of its edges ($\mathbf{E}_i$)	Align the agent's direction ($\mathbf{D}_j$) along the object's edge	Choose $\mathbf{E}_i$ if $\mathbf{D}_j.\mathbf{E}_i$ is positive; else choose $-\mathbf{E}_i$
	With a corner (two edges)	Align the agent along the angular bisector of the two edges	The agent moves along this angular bisector, away from the object
	Head on collision when the object moves towards the agent	Align the agent along the perpendicular of the intersecting edge	Translate the object along this perpendicular, away from the object, by 2 d or 3d; d is the diameter of the agent's envelop
Collision with other agents	The agent's direction $\mathbf{D}_j$ is altered by $\pm\alpha$. The agent is projected by distance d along $\mathbf{D}_j \pm \alpha$ and check for collision is performed	If either path is clear, that is chosen and the agent moves forward	If the path is not clear, α is increased by a factor of 1.25 and procedure (previous two columns) is repeated

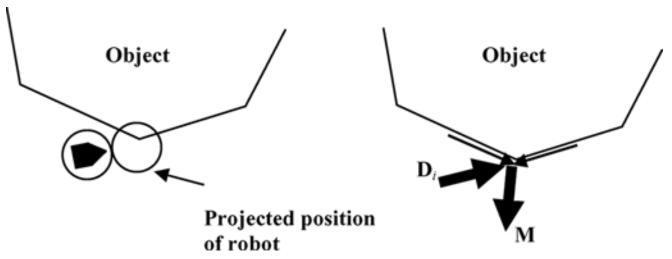

Fig. 14. Case of intersection with multiple (two) edges is shown. Robot is finally directed along direction M pointing outward.

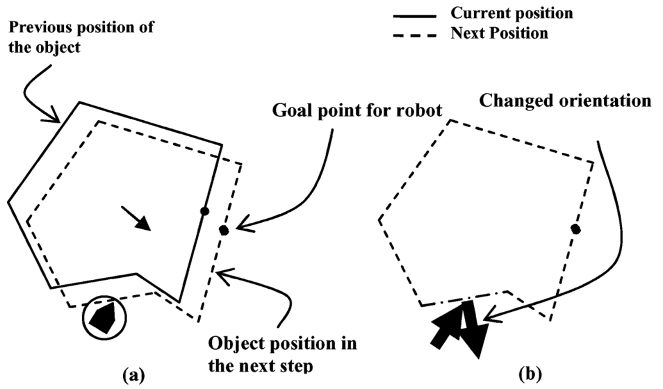

Fig. 15. When a robot is found intersecting/inside the next position of the object, it is directed toward the outward normal. (a) shows the robot intersecting with projected object position. (b) shows the direction along which the robot is directed.

bisector of the two edges. Also, it is ensured that the modified robot orientation is pointing away from the object. Case of multiple edge intersection is considered separately because the first case (intersection with a single edge) may result in a direction which will have the robot moving within the object's polygonal boundary.

3.4. *Avoiding head-on collision with the object*

When a robot faces a situation in which the moving object is approaching toward the robot, a separate technique is required because orienting the robot with the intersecting edge is not appropriate in this case. The robot should move away if its goal point is not there on that edge. In each step of the algorithm, it is checked whether a robot is entering inside the object polygon (or convex envelope) or intersecting with it. If found so, the robot's orientation is directed along the line perpendicular to the intersecting edge in outward direction of the object (envelope). Also robot's position is changed by projecting it along the new direction by an amount *2d* or *3d*, where *d* is the diameter of the envelope of the robot. The distance by which the robot should be moved depends on the velocity of the object towards the robot. The aim of this step is to keep the robot away from the edge so that it can also carry on moving toward its goal point (Fig. 15).

3.5. *Collision avoidance among robots*

For a numbers of multiagents participating in capture, collision among them is avoided by the following set of rules. Out of the total *n* robots, it is first determined which robots are going to intersect with other robots by projecting their envelopes with a specified distance *d* (see Fig. 16). This distance *d is* generally taken as less than or equal to the diameter of robot envelope to ensure that no intersection occurs at the intermediate positions.

If a robot is found intersecting, the direction of projection is altered by specified angle α(15°, say). That is, the robot's image is projected by adding or subtracting (picking the sign randomly) α from its current orientation, and then, again checking for the intersection with other robots' projection. If the path is clear, the robot moves ahead to the new projected position.

In case the path is not clear, the alternate sign of α is chosen, and collision (intersection) check is repeated again. Even after this try, if the path does not get cleared, the value of α is increased by factor (say, 1.25). The conical search

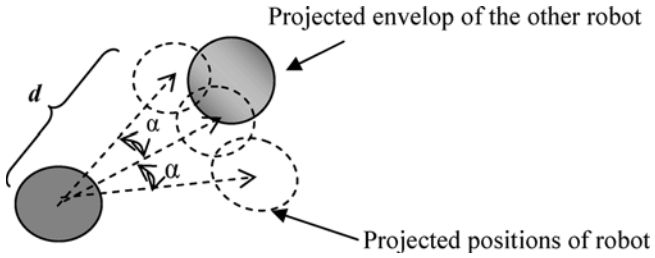
Projected envelop of the other robot
d
α
α
Projected positions of robot

Fig. 16. Robot image is projected and its intersection is checked with the other projected robots. If it is found intersecting, then projection is altered in both directions by certain angle.

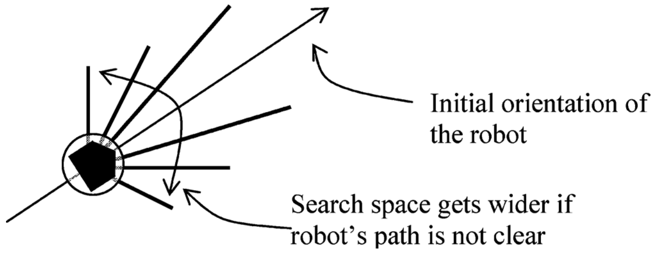
Initial orientation of the robot
Search space gets wider if robot's path is not clear

Fig. 17. It shows how the conical search space widens if the robot does not find a clear path when robot–robot collision is considered.

space becomes wider (see Fig. 17) and the robot again tries to achieve a clear path with the aforementioned method. This procedure is repeated until a robot gets a cleared path. This method is also applicable when more than two robots intersect at a time. Even when projected images of many robots tend to intersect with each other, this method gives a clear path for the robot if it exists at the cost of little increase in computational expense.

4. Guiding the Object After Catching

In previous sections, algorithms to determine the optimum goal points on the moving object and then to approach those goal points using projective path planning are discussed. We consider the object to be alive before it is arrested. The object is then assumed dead, i.e., it will not escape again on its own even when form closure is retracted. After the object is captured, it is required to be transported to a desired location.

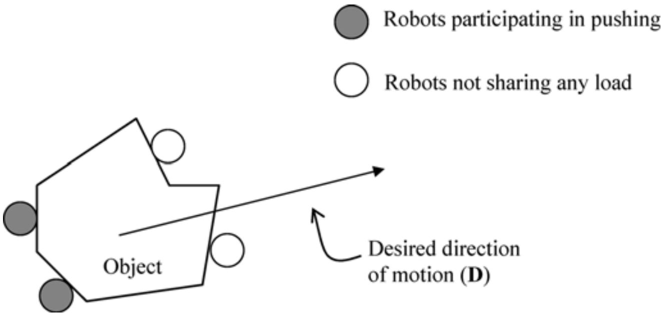
Robots participating in pushing
Robots not sharing any load
Object
Desired direction of motion (D)

Fig. 18. Object is taken to a desired location after catching it by four mobile robots in form closure configuration. Robots shown as filled circles are applying the force for pushing while robots shown as blank circles are not applying any load.

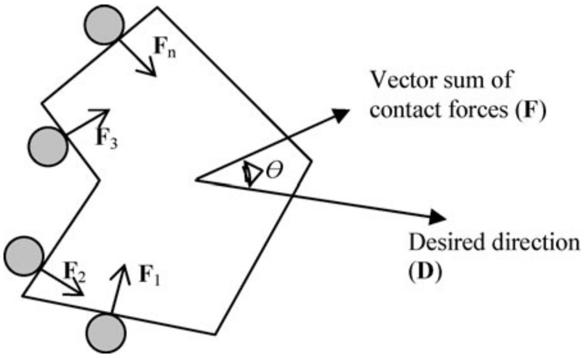

Fig. 19. D is the direction along which the object is to be transported; F is the vector sum of forces applied by multiagents on the boundary.

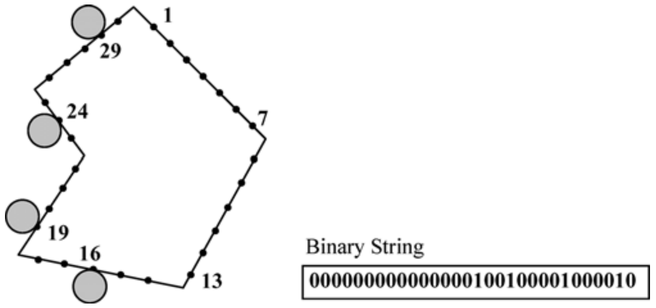

Fig. 20. Binary string is used as the decision variable in GA. Presence of 1 in the string represents robot at that point and 0 represents no robot.

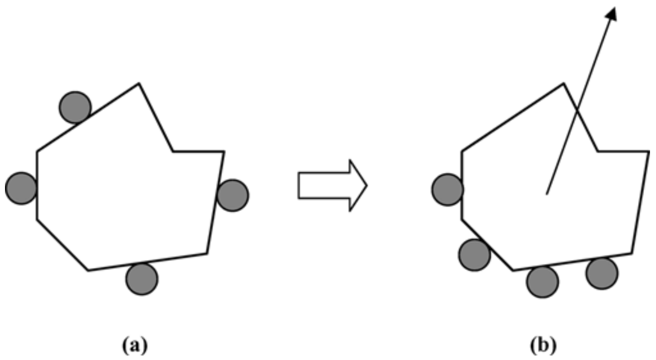

Fig. 21 (a) Robots catch the moving object and bring it to halt. (b) Robots reposition themselves to push the object toward the desired location. The thin arrow shows the desired direction along which the object is to be transported.

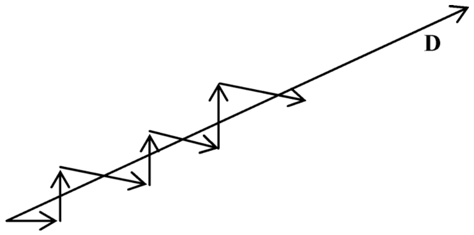

Fig. 22. Broken serpentine motion of the CG of the object is shown around the primary direction of motion when active–passive modes are switched between the two groups of multiagents.

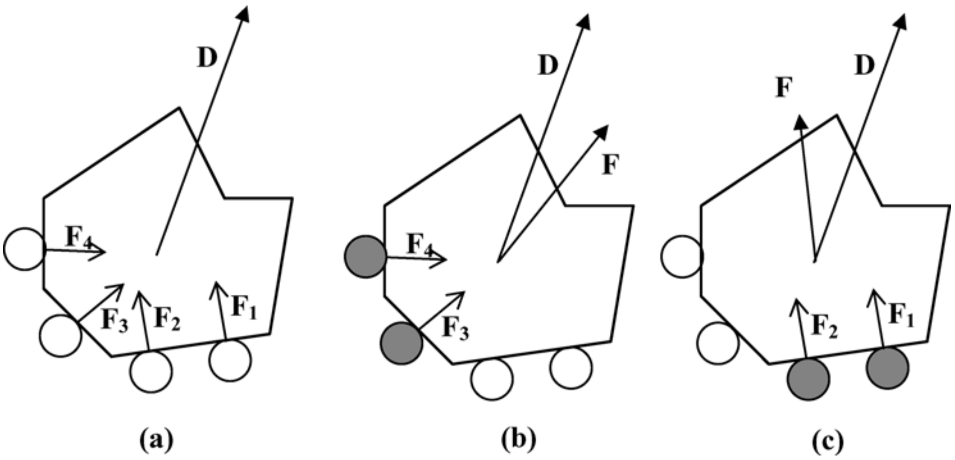

Fig. 23. Filled circles represent robots in active-pushing mode fazing to push the object and blank circles represent robots in passive mode.

Form closure arrangement of the multiagents is capable of transporting the object, if all robots move in synchronization. The need for reorienting the robots arises because the agents that are ahead of the object do not contribute in pushing the object to the desired location; they are just moving ahead of the object. As the object is assumed to be compromised post capture, there is no need to guard it against escape from the front side (Fig. 18).

4.1. Optimization in reorientation

To utilize all robots for pushing in a better manner, some optimal reorientation of robots is required (Fig. 19). This would ensure that the pushing force would be shared among the multiagents. The objective is to maximize the resultant force from the multiagents along the unit vector **D**, which is the desired direction of motion of the object from its arrested position. We consider that robots are applying unit force in the direction of the normal at the respective contact points on the boundary. The objective function is constructed as

$$\text{Maximize} \quad \begin{aligned} f &= \mathbf{D}.(\mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \cdots\cdots + \mathbf{F}_n) \\ f &= \mathbf{D}.\mathbf{F} \end{aligned} \tag{5}$$

where $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n$ are the unit forces applied by the robots on the object boundary at their contact points. The result of maximizing the dot product between the vectors **D** and **F** will eventually minimize the angle θ between them (see Fig. 19). A constraint in this optimization problem is the total number of agents pushing the object (n), which would be the same as the number that was used in catching the object while it was trying to escape. The design variables are the robots' positions around the object boundary, which is discretized into a finite set of n points. A vector in the population is composed of n entries, each of which can either have 0 or 1 values. An entry having a value of 0 implies that the robot is not present at that point while that of 1 suggests the presence of a robot (Fig. 20).

4.2. Repositioning of agents post capture

After determining the optimal pushing configuration of mobile robots, the latter are required to be repositioned (Fig. 21). The robots again adopt the projective path planning

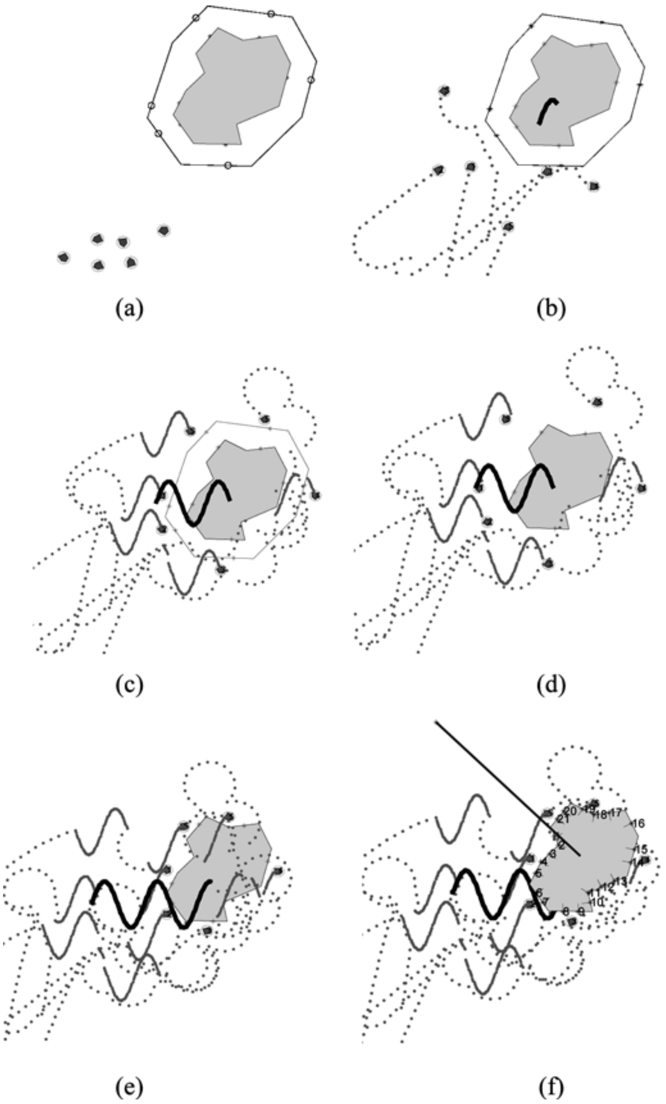

Fig. 24. This example shows the simulation result for an arbitrary-shaped polygon moving on a sine curve. Object's CG's path is shown by thick black curve. (a)–(c) show robots tracing the goal points on the envelope. (d) and (e) show robots tracing the goal points on the object. (f) shows the desired direction where the object is to be carried after its compromise.

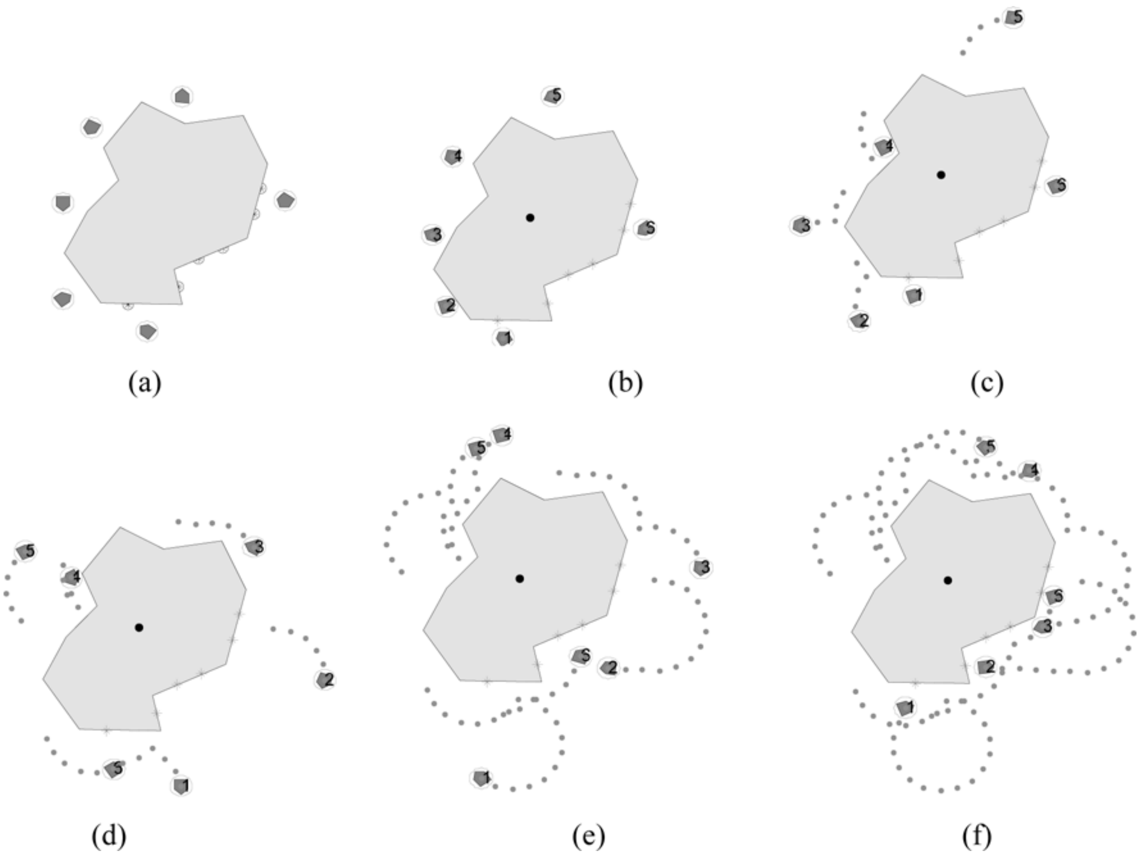

Fig. 25. Reorientation of robots from optimal catch configuration to optimal pushing configuration (a)–(f). '*' on the object boundary show the optimal configuration for pushing along desired direction. Dotted lines show the paths of the robots. The object is stationary after it is compromised.

method to reach their new goal points. When all robots reach their new positions, they can all start pushing the object. We assume that the object is dragged against friction and the resistance force is constant. If all robots now apply their maximum possible normal forces on the object, the latter may not always move toward the desired direction. The reason is that the resultant force vector **F** will have one component perpendicular to the vector **D**. This component will cause the object to deviate from the desired path.

4.3. Serpentine motion

A method is suggested to guide the object to the desired location in a serpentine like motion. This process involves switching on and off the robots such that the primary direction of motion **D** is broken into a sequence of motion along two directions as shown in Fig. 22. The robots are first divided into two groups. One group remains active in pushing at a time while the other remains passive; after each step, the groups switch their role. The active group members push the object at their respective goal points, while the passive group members follow their goal points when the object moves forward. This process results in a broken serpentine motion along the desired line of travel.

When the deviation of the object CG from direction **D** exceeds the predetermined distance, the modes are switched so that the CG can again be moved to the other side of the desired line. This predetermined distance can be decreased as the object moves toward its destination. The robots stop pushing the object once it is desirably close to its final goal point. Switching of modes from one to another is shown in Fig. 23.

5. Results

The simulation result for the capture of an arbitrary-shaped prismatic object moving along a sinusoidal curve is presented. These figures show robots capturing (Fig. 24), reorienting (Fig. 25), and transporting (Fig. 26) the object to desired location. Fig. 24 explains how the agents trace the object using projective path planning algorithm after the optimal contact points have been found. Fig. 24(a) shows the initial position of robots and object along with the convex envelope around the object. The '*' marks on the object boundary and on the convex envelope surrounding it show the corresponding optimized goal points for robots. Robots start tracing their respective goal points on the convex envelope, first using projective path planning algorithm (Fig. 24(b) and (c)). When all the robots reach their goal points on envelope, they simultaneously start tracking final goal points on the object boundary (Fig. 24(d) and (e)). Fig. 24(f) shows a line representing the direction along which the object is to be carried after it is compromised in form closure by the mobile robots.

Figures 25(a)–(d) show reorientation of robots from the optimized catch configuration to the optimized pushing configuration. The optimized pushing configuration is

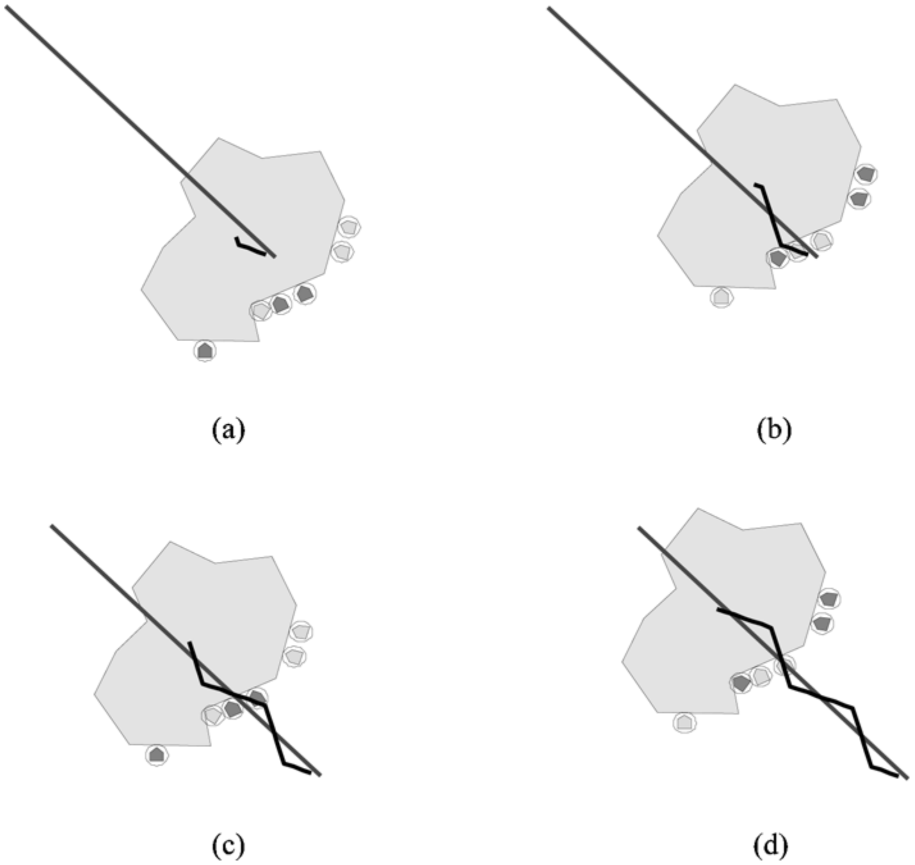

Fig. 26. It shows transporting the object to desired location in a broken serpentine motion. Robots filled with red/dark represent the active pushing mode and robots in green are in passive modes. Straight blue line represents the desired direction of transportation and the black zigzag line shows the path of the object's CG.

calculated based on the direction of desired object motion shown by a thin line using the algorithm in Section 4.1. The parameters used in genetic algorithm for the new optimization function are same as the previous optimization when determining optimal grasp points. The agents start moving to their new positions using projective path planning algorithm.

After achieving optimized pushing configuration, robots get divided into two groups as explained in Section 4.3. At any instant, only one group pushes the object while the other group simply moves along with the object (robots in red/dark push the object while in green/light are idle). This action results in broken serpentine motion of the dead object, which is shown by the zigzag path of object's CG (Fig. 26(a)–(d)) toward the goal point; when the object's CG is close to the goal point (within given tolerance limit), the simulation stops.

6. Conclusion and Future Work

The main contribution of this paper is that the best positions for the capture and the number of robots for capturing a gradually moving object is found using genetic algorithm. The concept of accessibility angle is used to find the goal points such that form closure is satisfied. The newly developed projective path planning scheme, which is decentralized and collision free, moves the robots to the respective goal points, and constrains the object. Finally, the compromised object is optimally moved to its desired goal position by repositioning the robots, and then, switching them as active or passive. The repositioning goal points are also determined optimally using genetic algorithm. Simulation results illustrate the effectiveness of the proposed methods. Future work is to implement this algorithm on the hardware. The path of the robot is to be smoothened in real time using Ferguson curves, using end slopes, and position information. Stationary obstacles will also be modeled using suitable methods.

References

1. A. Bicchi and V. Kumar "Robot Grasping and Contact: A Review," *Proceedings of IEEE International Conference on Robotics and Automation*, (2000), pp. 348–353.
2. T. Yoshikawa, "Passive and Active Closures by Constraining Mechanisms," *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota (Apr. 1996).
3. E. Rimon and A. Blake, "Caging 2D Bodies by 1-Parameter Two-Fingered Gripping Systems," *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota (Apr. 1996).
4. C. Davidson and A. Blake, "Caging Planar Objects with a Three-Finger One-Parameter Gripper," *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, (May 1998).
5. D. Ding, Y. Liu, J. Zhang and A. Knoll, "Computation of Fingertip Positions for a Form-Closure Grasp," *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul, Korea. (May 21–26, 2001).

6. J. Ponce and B. Faverjon, "On computing three-finger force closure grasp of polygonal objects," *IEEE Trans. Robot. Autom.*, **11**(6), 868–889 (Dec. 1995).
7. M. Kaneko, K. Harada and T. Tsuji, "Dynamic Friction Closure," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC (May 2002).
8. Y. Aiyama, Ma. Hara, T. Yabuki, J. Ota and T. Arai, "Cooperative transport by two four legged robot by implicit communication," *Robot. Autonom. Sys.* **29**, 31–19 (1999).
9. M. Hashimoto, F. Oba and S. Zenitani, "Object-Transportation Control by Multiple Wheeled Vehicle-Planar Cartesian Manipulator Systems," *Proceedings of IEEE International Conference on Robotics and Automation*, (1995) pp. 2267–2272.
10. M. Hashimoto and F. A. Uminoroib, "Dynamic Control Approach for Motion Coordination of Multiple Wheeled Mobile Robots Transporting a Single Object," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan (Jul. 26–30, 1993).
11. M. N. Ahmadabadi and E. Nakano, "A constrain and move approach to distributed object manipulation," *IEEE Trans. Robot. Autom.* **17**(2), 157–172 (Apr. 2001).
12. S. Yamada and J. Saito, "Adaptive action selection without explicit communication for multi-robot box-pushing," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.* **31**(3), 398–404 (Aug. 2001).
13. D. Sun and J. K. Mills, "Manipulating rigid payloads with multiple robots using compliant grippers," *IEEE/ASME Trans. Mechatronics* **7**(1), 23–34 (Mar. 2002).
14. C. Galta, R. Lumia, J. Wood and G. Starr, "An Efficient Method to Compute Three Fingered Planar Object Grasps Using Active Contour Models," *Proceedings of IEEE/RSJ International conference on Intelligent Rebote and Systems* (Sep. 2004) pp. 3674–3679.
15. C. Belta and V. Kumar, "Abstraction and control for group of robots," *IEEE Trans. Robot.* **20**(5), 865–875 (Oct. 2004).
16. W. Burgard, M. Moors, C. Stachniss and F. E. Schneider "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, **21**(3), 367–378 (Jun. 2005).
17. T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.* **18**(1), 94–103 (Feb. 2002).
18. J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar and J. P. Ostrowski, "Cooperative Localization and Control for Multi-Robot Manipulation," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2001) pp. 631–636.
19. S. Carpin and L. E. Parker, "Cooperative Leader Following in a Distributed Multi-Robot System," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC (May 2002) pp. 2994–3001.
20. R. Fierro, A. K. Das, V. Kumar and J. P. Ostrowski, "Hybrid Control of Formations of Robots," *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul, Korea, (May 21–26, 2001) pp. 157–162.
21. R. Fierro and A. K. Das, "Hybrid Control of Reconfigurable Robot Formations," *Proceedings of the American Control Conference*, Denver, Colorado, (Jun 4–6, 2003) pp. 4607–4612.
22. A. Sudsang and J. Ponce, "A New Approach to Motion Planning for Disc-Shaped Robots Manipulating a Polygonal Object in the Plane," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, CA, (Apr. 2000) pp. 1068–1075.
23. Z. Wang and V. Kumar, "Object Closure and Manipulation by Multiple Cooperating Mobile Robots," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, (May 2002) pp. 394–399.
24. G. C. Buttazzo, B. Allotta and F. P. Fanizza, "Mousebuster: A robot for real-time catching," *IEEE Control Syst. Mag.* **14**(1), 49–56 (Feb. 1994).
25. Y. Liu, A. Hoover and I. Walker, "A timing model for vision-based control of industrial robots", *IEEE Transactions on robotics* **20**(5), 2004, pp. 891–898.
26. P. Song and V. Kumar, "A Potential Field Based Approach to Multi-Robot Manipulation," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, (May 2002) pp. 1217–1222.
27. H. Asada and A. B. By, "Kinematic analysis of work-part fixturing for flexible assembly with automatically reconfigurable fixtures," *IEEE J. Robot. Autom.* **Ra-1**(2), 86–93 (1985).
28. M. Gordy, *A Matlab Routine for Function Maximization Using Genetic Algorithm*. (Matlab Codes. GA, 1996).
29. P. Sharma, A. Dutta and A. Saxena "Determination of Optimal Contact Points for Constraining a Prismatic Object by a Group of Mobile Robots," *Proceedings of the IEEE Conference on Robotics and Mechatronics*, RAM, Bangkok, (2006) pp. 86–90.
30. P. Sharma, A. Saxena and A. Dutta, "Multi Agent Form Closure Capture of a Generic 2D Polygonal Object Based on Projective Path Planning," *Proceedings of the ASME 2006 International Design Engineering Technical Conferences*, Philadelphia, USA, (Sep., 2006) pp. 1–8.