Task Allocation and Collision free path planning of centralized multi-robots system for industrial inspection using heuristic methods

•••

A Summary

Amrit Kochar
2014B4A70821P

The Problem

- Task allocation and **collision-free path planning** for 3 robots in a common workplace (90 tasks).
- The grid map of the inspection plant is stored as a matrix.
- Task allocation is done by using a GA.
- Path planning is carried out for each robot by implementing the A* algorithm.

Assumptions

- Each robot can execute only one task at a time.
- Only one robot is required to execute each task.
- Each task is executed only once.
- All the tasks are to be executed.
- All the robots start from the depots at the same time.

Mathematical Model

- The first two robots are randomly assigned (25-35) number of tasks and the remaining is assigned to the third one. *(Random number generation)*
- Instead of 90! Combinations with a single robot, we now have 90*a*b combinations with 3 robots.

Let us consider an optimization problem, where a group of m robots $R = \{R_1, R_2, \ldots, R_i, \ldots, R_m\}$ are to be assigned inspection task at n locations $T = \{T_1, T_2, \ldots, T_j, \ldots, T_n\}$ in an optimal sense, such that either the completion time (that is, maximum of the traveling times taken by m robots) or total fuel consumption (representative of total traveled distance) becomes the minimum. Let C_{ij} be the cost (expressed in terms of the sum of traveling time and inspection time) of R_i robot for inspecting at T_j location. Therefore, C_{ij} can be expressed as follows:

$$C_{ij} = C_{ijv} + C_{ijs}, \tag{1}$$

where C_{ijv} represents the traveling time and C_{ijs} denotes the inspection time. It is to be noted that C_{ijv} is calculated by knowing the distance between ith robot (R_i) and jth location (T_j), that is, d_{ij}, velocity of the robot v_i and coefficient of ground condition $\alpha(1, 10)$ as follows:

$$C_{ijv} = \alpha\left(\frac{d_{ij}}{v_i}\right). \tag{2}$$

Therefore, completion time (CT) can be expressed as follows:

$$CT = \max\left\{\sum_{j=1}^{n} C_{1j}, \ldots, \sum_{j=1}^{n} C_{ij}, \ldots, \sum_{j=1}^{n} C_{mj}, \ldots,\right\}, \tag{3}$$

where $C_{ij} > 0$ if task T_j is allocated to R_j; otherwise, it is set equal to 0.

Now, total fuel consumption (FC) (representative of total distance traveled) can be obtained as follows:

$$FC = \sum_{i=1}^{m}\sum_{j=1}^{n} \delta_i C_{ij}, \tag{4}$$

where δ_i is the coefficient of fuel consumption, which has been assumed to be the same for all the robots.

Environment Representation

- The map is represented as a grid and stored as a matrix in memory.
- The accessible areas are denoted by 0 and the inaccessible ones by 1.

Solution Representation

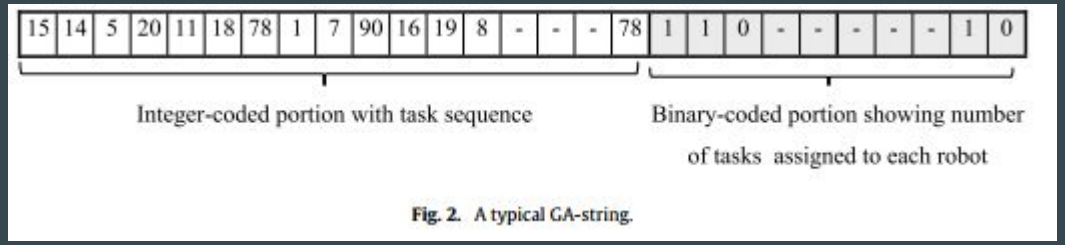

Fig. 2. A typical GA-string.

- The first part of the string consists of randomly generated numbers, which denote the sequence of tasks. *(a permutation of all tasks at hand)*
- The binary part of the string represents the number of tasks allocated to first two robots. 9 bits are used for this purpose.
- Number of tasks for the third robot is calculated, 90-a-b.

Greedy Initialization

- Used to group nearby tasks together.
- A starting point is selected and then in a loop the nearest unselected location is added to the sequence.

```
function Greedy (random Point)
 String (1) = random Point
 Set U contains all points from 1 to loc
for j = 2 … loc
String(j) =  Nearest Point from string (j-1) in set U
delete String(j) from U
end for
return String
end function
```

Fitness Evaluation

Case 1: Minimization of Total Fuel Consumption

$$f_i = \frac{1}{sum\left\{\sum_{j=1}^{n} C_{1j}, .., \sum_{j=1}^{n} C_{ij} \cdots \sum_{j=1}^{n} C_{mj}\right\}}.$$

Case 2: Minimization of Total Completion Time

$$f_i = \frac{1}{\max\left\{\sum_{j=1}^{n} C_{1j}, .., \sum_{j=1}^{n} C_{ij} \cdots \sum_{j=1}^{n} C_{mj}\right\}}.$$

Collision Checking

```
Find total path travelled by all robots R1,R2,R3..., etc.
collision = 0
while (collision  = = 1) or (check = =1 )
for m= (1 to number of robots)
for n= (1 to number of robots)
for i = 1 to (min{ length of Rm, length of Rn})
if  { Rm (i, 1 to 2)} = =  { Rn (i, 1 to 2 )}
        collision = 1
end if
end for
end for
end for
check = 1
end while
return collision
```

Let us take a hypothetical situation, where the position coordinates of all three robots at different time steps are given below. At time $t = 5$ s, the coordinates of R_2 and R_3 were (20, 40), which means that they would collide at that point. This was estimated and the solution having the chance of collision was penalized.

Path coordinates of Robots							
	R1		*R2*		*R3*		
Time(s)	*x*	*y*	*x*	*y*	*x*	*y*	
1	18	44	20	44	22	44	
2	18	43	20	43	22	43	
3	17	43	20	42	22	42	
4	17	42	20	41	22	41	
5	16	42	20	40	20	40	→ Collision
6	16	41	20	39	23	40	
7	15	41	21	39	24	40	
8	15	40	21	38	25	40	
9	14	40	21	37	26	40	
10	14	39	21	36	27	40	
11	13	39	21	35	28	40	
12	12	39	21	34	28	39	

- Solutions with collision were heavily penalised so that they have a low probability to occur in future generations.

GA operations

Reproduction

- Ranking scheme is used,eg. lowest FF value is rank 1.

Crossover

- For the first part of solution string, we use Partially mapped Crossover. In this, substrings are exchanged and then the new strings are legalized by considering the mapping.
- For the second part, uniform crossover was performed. Each bit position was associated with a randomly generated number and depending upon whether it is <,>,= 0.5 crossover took place.

GA operations (Continued)

Parent String	1	3	4	5	6	7	8	2	9	10
Mutated String	1	3	8	7	6	5	4	2	9	10

Mutation

- For the first part of the string, two random numbers lying between 1 and string length is generated and a flip mutation is implemented.
- For the second part, a bit-wise probabilistic mutation was implemented. Each bit position was associated with a randomly generated number and depending upon whether it is <,>,= 0.5 mutation took place.

Some Probabilities for the operations

p_{c1}: Probability of Partially-Mapped Crossover used for the integer-coded portion of the string,

p_{c2}: Probability of Uniform Crossover utilized for the binary-coded portion of the string,

p_{m1}: Probability of Mutation used for the integer-coded portion of the string,

p_{m2}: Probability of Mutation utilized for the binary-coded portion of the string,

Gr: Percentage of Greedy optimized chromosomes for Initialization,

N: Population size, and

G_{Max}: Maximum number of generations.

Results

Case 1 Minimization of total fuel consumption without greedy initialization and without collision avoidance

Fig. 3 displays the results of parametric study. The following parameters were found to yield the best results: $p_{c1} = 0.75$, $p_{c2} = 0.8$, $p_{m1} = 0.09$, $p_{m2} = 0.05$, $N = 500$, $G_{max} = 2000$. It is important to mention that in this case, neither greedy initialization nor collision avoidance had been considered. In order to complete inspection, total fuel consumption and completion time were obtained as 528 units and 230 s, respectively. Fig. 4 shows the path traced/traveled by three robots to complete the inspection task.

Case 2—Minimization of total fuel consumption with greedy initialization and without collision avoidance

In this case, the method of greedy initialization was adopted in the population of solutions. However, collision avoidance scheme was not used. A thorough GA-parametric study had been conducted and the following parameters were seen to give the best results: $p_{c1} = 0.85$, $p_{c2} = 0.95$, $p_{m1} = 0.08$, $p_{m2} = 0.04$, $Gr = 61\%$, $N = 500$, $G_{max} = 800$.

The three robots were able to perform the inspection task by consuming fuel of 412 units and the task completion time was found to be equal to 154 s.

- Greedy initialization improved results significantly and hence, it was adopted case 2 onwards.

Results (Continued)

Case 3—Minimization of total fuel consumption with greedy initialization and collision avoidance

In this case also, an attempt was made to minimize total fuel consumption. Besides greedy initialization, a collision avoidance scheme (as discussed above) had been implemented to ensure collision-free navigation of the inspecting robots. The following GA-parameters were found to yield the best results: $p_{c1} = 0.85$, $p_{c2} = 0.8$, $p_{m1} = 0.2$, $p_{m2} = 0.02$, $Gr = 30\%$, $N = 700$, $G_{max} = 300$. Fig. 6 displays the paths followed by the three inspecting robots.

Total fuel consumption and completion time were obtained as 454 units and 172 s, respectively. To ensure collision-free navigation, the inspecting robots in this case were seen to take more fuel and completion time in comparison with Case 2. However, Case 3 deals with a more practical problem compared to that of Case 2.

Case 4—Minimization of task completion time with greedy initialization and collision avoidance

In this case, task completion time of the inspecting robots had been minimized after considering the chance of their collisions. A greedy initialization had been utilized in the algorithm, as discussed above. The following GA-parameters had yielded the best results: $p_{c1} = 0.85$, $p_{c2} = 0.9$, $p_{m1} = 0.07$, $p_{m2} = 0.06$, $Gr = 50\%$, $N = 800$, $G_{max} = 900$.

Fig. 7 shows the paths of three robots. The robots were able to complete the inspection in 156 s by consuming fuel of 462 units.

The results obtained in this study were found to be better compared to those of the previous work [11]. It might have happened due to the reasons that the GA not only determined the optimal schedule but also assigned the optimized number of tasks to each robot. With the greedy initialization, better results were observed. In the collision avoidance scheme, the directions of motion of the robot were restricted to four only. This increased the fuel consumption and task completion time.

Conclusion

- Better results are observed because the algorithm not only optimizes the schedule but also the number of tasks for each robot.
- Re-routing is efficient due to the one time task of computing distance matrix.
- We can encode a lot more of real world conditions.
- Using modern optimization techniques should help.