Reactive Behaviour Switch in a Multi-robot application generating single-robot and multi-robot solutions

Francesco Mauro

Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, Delft, the Netherlands

Abstract

In recent years, the postal market has experienced an exponential growth of parcel volumes. So far, this growing demand has been handled through an amplification of the supply capacity and the purchase of new sorting equipment. Clearly, this solution is neither sustainable nor future-proof. However, the ability of postal operators to plan alternative strategies is constrained by the innermost characteristics of conventional sorting machines, which do not provide enough flexibility to handle the uncertainties in the postal market. Artificial intelligence and robotics are deemed the next game-changers in the field of logistics. A multi-robot parcel sorting system is able to provide flexibility and scalability desired by postal operators. However, within this application domain, robots need to handle concurrently ST-SR-IA (i.e. transport of small loads) and ST-MR-IA tasks (i.e. transport of big loads). In this research, we propose an algorithm, in which robots switch reactively their behaviours when facing different tasks. This algorithm is simple, efficient and scalable. However, the integration of this solution with leader-follower algorithm leads to reducing the fault tolerance of the sorting system, as shown in the experimental designs. To solve this problem, we propose an assistance mechanism that aims at setting free the trapped robots in formations.

Keywords: Multi-robot systems, Warehouse automation, Multi-robot task assignment, Agent-based simulation, Multi-robot motion coordination

1. Introduction

In the postal industry, little has changed in the last decades with regard to the automation systems used for parcel sorting. The conventional automated sorting systems mostly operated today comprise fixed and large machinery (Yunardi, 2015).

Although the widespread use of these traditional systems, they constrain the ability of industries to plan short- and long-term strategies as a result of their inflexibility.

The future of warehouse systems sets upon their flexibility.

Warehouse systems do not have to be adjustable to just a set of pre-defined scenarios, but they have to be able to cope with unpredictable circumstances.

Recently, multi autonomous mobile devices have emerged as transportation means in warehouses. These systems have changed the way we look at warehouse automation, as they have altered our vision of warehouses from static networks of fixed machineries to distributed networks of autonomous agents.

Multi-robot parcel-sorting systems can provide postal operators with high flexibility and scalability to defeat the fluctuation of parcel volumes.

However, within this domain of application, robots need to transport light and low volume (ST-SR-IA) and heavy and high volume (ST-MR-IA) parcels. An important constraint is that robots should be homogenous and of limited dimensions. This problem represents the main objective of this scientific paper. In light of this objective, the main research question can be formulated as:

"How can homogenous robots perform concurrently ST-SR-IA and ST-MR-IA within the same application?"

Furthermore, in this article, we also intend assessing *the impact of cooperative behaviour on fault tolerance*. These research objectives are answered by following a structured approach.

In Section 2, the article explores previous researches on this topic. In Section 3, we illustrate the steps followed to develop a conceptual model of a multi-robot parcel sorting system. In Section 4, we detail the reactive algorithm used to address ST-SR-IA and ST-MR-IA tasks using homogenous robots. In Section 5, we discuss the simulation environment where we develop experiments and collect results.

In Section 6, we display the experimental designs and results obtained from simulations. Finally, in Sections 7 and 8, we infer conclusions and suggest the next steps for the scientific research, respectively.

2. Previous work

Multi-robot task allocation is one of the most challenging and investigated domains of multi-robot systems (MRS), which deals with the way robots are assigned to the tasks in such a way that the system performance is optimized and constraints are satisfied.

Gerkey and Mataric (2004) propose a taxonomy that is useful in order to distinguish the type of algorithms that can be used for different categories of MRS. They describe multi-robot task allocation problems based on three determinants: single-task (ST) vs multi-task (MT); single-robot (SR) vs multi-robot (MR); and instantaneous assignment (IA) vs time-extended assignment (TA). In this case, we only consider problems with instantaneous assignment, since time-extended assignment problems are more related to scheduling problems than to assignment problems.

ST-SR-IA are the most well-known and simplest problems, where: each robot is able to perform only a single task at a time; each task only requires one robot to be accomplished; and the allocation of the tasks to the robots is instantaneous.

A vast literature exists on solving these problems, with MIP algorithms or auction-based algorithms typically leveraged to solve them (Lattarulo and Parks, 2012; Khamis, 2015).

ST-MR-IA are problems where each robot is able to perform only a single task at a time, but each task requires the combined effort of multiple robots. These problems present very complex task decomposition (NP-hard) and only few researches have attempted to solve these task assignment problems. The most employed algorithms to cope with them are the Set Partitioning Problem (SPP), which divide a set of robots into finite subsets of feasible teams, each of which tries to optimize its own utility. However, as underlined in Sheohory and Kraus (1995), the SPP solutions have two main deficiencies, namely (1) the computation complexity is exponential in the number of robots, (2) these solutions are centralized. To address these problems, the authors propose a distributed set-partitioning algorithm, with agents calculating and forming coalitions without the assistance of a central agent. Indeed, agents create coalitions based on their capabilities and the requirements of the tasks. The main disadvantage of this algorithm is that the average computational complexity is high and the solution does not scale well with increasing number of agents. However, this solution decreases the inter-agent communication. This algorithm can be also used for applications with concurrent execution of ST-SR-IA and ST-MR-IA tasks.

Parker and Tang (2006) develop an algorithm, termed ASyMTRe to address the ST-MR-IA multi-robot task allocation problem. The objective of this algorithm is to solve ST-MR-IA problems by forming coalitions, i.e. by organizing multiple robots into subgroups to accomplish a given tasks. The collaboration among robots is achieved by using robot schemas. Each robot contains a schema, i.e. a control framework that includes inputs, outputs, local variables, behaviours. Robot schemas define how the input needs to be processed in order to generate a certain output. The ASyMTRe algorithm builds a network of schemas connecting the outputs of one robot schema to the inputs of other robot schemas. In this way, computations from multiple schemas of robots are summed up and normalized to produce the desired cooperative behaviours. Furthermore, the authors develop a distributed version of this algorithm, termed ASyMTRe-D, which produces more reliable and flexible results, but it lacks of quality solutions.

In Tang and Parker (2007), the ASyMTRe-D algorithm is combined with a market-based task allocation algorithm. This combination is used to allow the robots to perform both ST-MR-IA and ST-SR-IA tasks within the same application. More specifically, the authors apply the ASyMTRe-D algorithm to solve ST-MR-IA tasks and auction-based algorithm to fulfil the ST-SR-IA tasks.

However, in their research, tasks are assigned sequentially in experiments. Therefore, at time x, task 1 is auctioneed, while at time x+1 and x+2, task 2 and 3 are auctioneed respectively. When the coalitions for these tasks are determined, other tasks are announced. This implies high amount of idleness for robots. Furthermore, considering the heterogeneity of robots, when the most capable robots are performing tasks, the less capable robots need to wait until the accomplishment of said tasks to form coalitions with capable robots.

Guerrero and Oliver (2012) propose another solution to address ST-MR-IA tasks. This task assignment problem is addressed using an auction-based algorithm in which the robot that discovers first a task becomes the leader and holds an auction to form coalitions of robots. In this algorithm, every task has a single leader that calls an auction in which the other members of the coalitions are decided based on their work capacities. The leader also decides the adequate group size for the execution of a task. This algorithm is able to decrease the computational and communication complexity of the task to address. However, in their research, the authors only focus on a solution for ST-MR-IA tasks. Furthermore, the authors do not investigate the disruptive situations the overreliance on leaders brings along.

Therefore, the performance of ST-SR-IA and ST-MR-IA tasks within the same domain of application, using homogenous robots and without sequential task assignment represents the knowledge gap for this study. In addition, an investigation and solution on the reduction of fault tolerance due to the overreliance on single robots (leaders) is necessary in this domain of application.

3. Model Conceptualization

Guided by the Design Science Research Methodology (K. Peffers et al., 2007), we design a model of a multi-robot parcel-sorting system.

As for actors starring in a drama, we first decide the main characters (system elements); next we assign them with a script (define how system elements are supposed to behave). These activities constitute the model conceptualization part, which can be seen as describing the narrative of the story.

3.1 System elements

The system elements of the multi-robot parcel-sorting system are *robots*, *parcels*, *pick-up buffers* (i.e. locations where robots obtain parcels) and *drop-off buffers* (i.e. destinations of parcels). Therefore, this system corresponds to a tuple (R, P, U, O), where:

$R = \{r_1, \ldots, r_i\}$ set of Robots

$P = \{p_1, \ldots, p_j\}$ set of Parcels

$U = \{u_1, \ldots, u_l\}$ set of Pick-Up Buffers

$O = \{o_1, \ldots, o_m\}$ set of Drop-Off Buffers

3.2 Robot process flowchart

The robot process flowchart embodies six main phases and the communication flow is also integrated in it (see Figure 1). As stated earlier, robots communicate in their environment with other robots (robot-to-robot), parcels (robot-to-parcel) and pick-up buffers (robot-to-pickup).

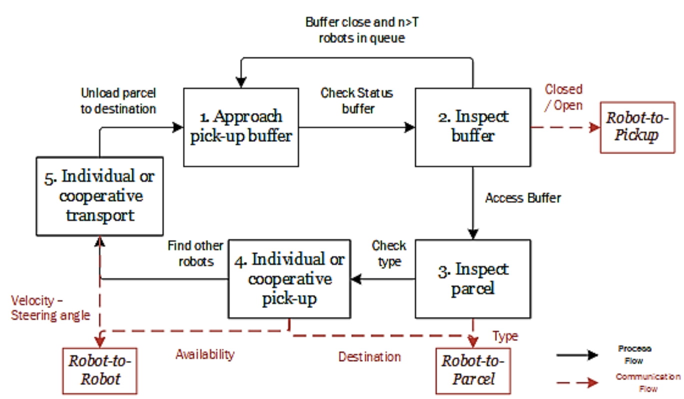

Figure 1: Robot Process Flowchart

Robots start operating from the queues of pick-up buffers, from where robots enter their assigned pick-up buffers on a first-in queue first-served basis. After robots deliver their parcels at the right containers, they need to re-calculate their pick-up buffers and move towards them.

Robot-to-pickup communication enables robots to acquire information regarding the open/closed state of the buffers. Closed buffers enforce robots to wait in queues before moving inside to collect parcels. This information-sharing allows controlling and regulating the incoming flow of robots into the buffers. Furthermore, the communication robot-to-pickup can be used to eliminate the possibility to disregard or overburden certain pick-up buffers.

Once inside a pick-up buffer, robots communicate with parcels to reserve them and eliminate the risk of deadlocks or situations where multiple robots argue for the same parcel. Indeed, when a robot claims a parcel, the couple (r_i, p_j) is formed and other robots can no longer request the same parcel. Robots require parcels to provide information regarding their types (e.g. weight, size). Based on the information obtained, robots decide whether they should act individually or collectively. If the parcel is light and low volume, we assume an individual robot is sufficient to complete this task. When an individual robot is not enough to load and transport a parcel, we assume four robots need to cooperate to complete this task. Loaded robots, i.e. robots with parcels, are ready to transport parcels to destination. This corresponds to a path-planning and collision avoidance problem, where robots need to find the best path to follow without collisions.

The transport of light parcels is a more elementary task, with robots finding the shortest path to transport the parcel to the right destination and, while traveling through their selected path, avoiding collisions with other robots. Once at destination, robots unload their parcels.

The cooperative transport is a less elementary task, since followers need to follow their leaders with a given distance d_i and a relative angle γ (lateral and longitudinal offset). A leader controls unidirectional its followers, frequently sending them its velocity and steering angle. Figure 2 shows the formation of the team of cooperative robots with UAVs, a leader and three followers. Every leader coordinates the actions of multiple robots, deciding the path to follow and avoiding obstacles. To avoid risks of collision, leaders exploit the sensing data of its follower to enlarge its vision.

Like the wagons of trains, followers convoy the leader with the objective of preserving the formation, with specific distance and angle.

The group moves preserving the formation until the destination is reached. Subsequently, the parcel is unloaded and the formation is dissolved.

It is important to underline that we are using the term robot generically. Therefore, in this application, robots can be intended as UAVs or UGVs.

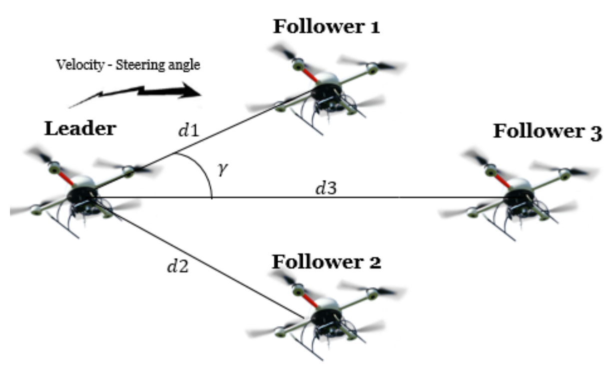

Figure 2: Leader-follower structure

4. Reactive algorithm for ST-SR-IA and ST-MR-IA tasks

Guerrero and Oliver (2012) address ST-MR-IA tasks, adopting a combination of leader-follower and auction-like algorithm. As shown in their research, this algorithm allows reducing inter-robot communication and computational complexity.

```
Reactive algorithm for execution of ST-SR-IA and ST-MR-IA tasks
for all r_i in R with load state = unloaded do
  leader = nobody
  load state = unloaded
  n followers = 0
  min robots = 3
  follower list = [min robots]
  if my parcel p_j = heavy and large then
    ;; cooperative behaviour
    leader = myself
    load state = loaded
    find r_i with load state = unloaded and my leader = nobody
    save r_i in temporary variable
    while nfollowers < min robots and found follower = false and found
    empty = false do
      if follower list [n followers] = r_i then
        found follower = true
      else
        if follower list [n followers] = empty then
          found empty = true
        else
          nfollowers =+ 1
        end if
      end if
    end while
    if found empty = true then
      follower list [n followers] = temporary variable
    end if
    for all followers in follower list do
      save my leader = myself
      save load state = loaded
    end for
  else
    ;; non-cooperative behaviour
    save load state = loaded
  end if
end for
for all r_i in R with load state = loaded do
  get my destination of my parcel
  save destination = destination of my parcel
  if my parcel = heavy and large and destination = o_m leftside or middleside
  then
    formation pattern1
  else
    formation pattern2
  end if
end for
```

Figure 3: Algorithm for reactive behaviour switch of robots

However, the authors have considered an application with only ST-MR-IA tasks. Therefore, the development of a solution for the concurrent performance of ST-SR-IA and ST-MR-IA tasks, within the same application without sequential announcement of tasks and using homogenous robots represents the main objective of this paper. In order to perform both types of task within a single application domain, we have developed a solution in which robots change dynamically their behaviours (or roles) when facing ST-SR-IA and ST-MR-IA.

By using this algorithm, when a robot discovers an ST-SR-IA task, it decides to act in a selfish (non-cooperative) manner, transporting parcels individually to destination.

```
Formation Pattern
if destination = o_m leftside or middleside then
  ;; formation pattern 1
  To follower 1:
  save steering angle = my steering angle
  save lat position = my lat position + l
  save long position = my long position
  To follower 2:
  save steering angle = my steering angle
  save lat position = my lat position
  save long position = my long position - l
  To follower 3:
  save steering angle = my steering angle
  save lat position = my lat position - l
  save long position = my long position - l
else
  ;; formation pattern 2
  To follower 1:
  save steering angle = my steering angle
  save lat position = my lat position - l
  save long position = my long position
  To follower 2:
  save steering angle = my steering angle
  save lat position = my lat position
  save long position = my long position + l
  To follower 3:
  save steering angle = my steering angle
  save lat position = my lat position - l
  save long position = my long position + l
end if
```

Figure 4: Formation patterns

While, when a robot discovers an ST-MR-IA task, it becomes a leader and starts auctioning to recruit followers based on their internal states. Therefore, adding to the previous research, we have developed a reactive algorithm for the

concurrent execution of ST-SR-IA and ST-MR-IA tasks (see Figure 3).
The term *reactive* refers to the reactive control approach (Sahota, 1994) in which sensors and actuators are tightly coupled to provide robots with the ability to react rapidly to changing environmental conditions (“stimulus-response”).

As shown, the reactive algorithm combines a leader-follower and auction-like algorithm for the performance of ST-MR-IA tasks, with robots dynamically changing roles into leaders and followers. The leader-follower strategy is also used for the creation of different formation patterns (see Figure 4) and for the motion coordination of robots with heavy and high volume parcels.

When coalitions of robots are created, leaders can assume different positions within the formations that give them the highest situational awareness.

A similar technique was used in Desai et al. (2001), to alter the shape of formations when facing diverse obstacles. In this application, leaders can position themselves on the front left or front right, dependently on the position of parcels destinations.
For the motion coordination of robots, two strategies could have been used, namely virtual structure and leader-follower. In this application, we have opted for a leader-follower algorithm. The leader-follower algorithm enables a substantial reduction of communication, since only one leader transmits commands to the other members. Additionally, as explained in Consolini et al. (2008) and in Mas and Kitts (2010), the leader-follower provides higher scalability compared to the virtual structure approach.
Introducing additional robots in a virtual structure composition affects the physics of the rigid body, thus decreasing the scalability of the algorithm.

The proposed algorithm allows the use of homogenous robots for the execution of ST-SR-IA and ST-MR-IA tasks. Furthermore, it reduces the average computational complexity, considering that only one agent decides upon the preferred coalitions.
However, the main problem related to the use of this algorithm is the overreliance on single agents. Consequently, when the leader or a follower of a coalition fails, disruptive situations arise.
Therefore, the cooperative behaviours of robots may decrease the robustness of the system, being the ability of the system to keep operating profitably even in the presence of partial failures.

Therefore, in the next paragraph, we evaluate the impact of cooperative behaviour on system fault tolerance.

5. Simulation and results

Once the conceptual model is developed, the next step is to implement it in an appropriate modelling environment.
Considering the large proportion of this system, where the actions of thousands of agents and objects are taken into consideration, agent-based simulation guarantees easy implementation, scalability, easy modification of simulation parameters and accurate description of agent/object classes.

Taking inspiration from the Alphabet Soup model (Hazard et al., 2006) and considering the layout of traditional sorting centres, we have designed the system displayed in Figure 5.

In this sorting environment, the entry gates, where the inbound trucks arrive, are located on the left side. Instead, the exit gates, where the sorted parcels are moved onto outbound trucks, are located on the bottom, top, and right sides. In this environment, there are 20 pick-up buffers (in green) and 50 drop-off buffers (in red). As can be observed, each pick-up buffer is connected to two queues, one entry queue and one exit queue.

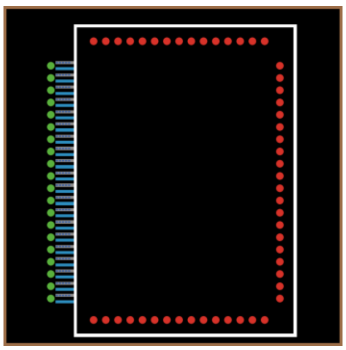

Figure 5: Simulation environment

Entry queues are those locations where robots wait to enter a pick-up buffer; while exit queues correspond to the lines robots drive through before entering the transport field. The transport field (in white) signals the area where robots transport parcels to the appropriate drop-off buffers.

6. Experimental designs

In the experimentation phase, computer experiments are conducted by altering the values of certain input parameters and inferences are deduced.

The objective of these experimental designs is to assess the impact of cooperative transport on system fault tolerance, i.e. the ability of a system to keep operating even in the presence of failure of one (or more) robots. Therefore, we want to design disruptive scenarios, where a number of robots fail and evaluate the impact on the system performance. In particular, we are interested in assessing the impact of robots facing ST-MR-IA tasks on fault tolerance.

In fact, when a robot fails during the transportation of these loads, the whole formation collapses, thus having a larger impact on fault tolerance in comparison to robots performing ST-SR-IA tasks.

Accordingly, the input parameters that we have decided to vary are only number of faulty robots (i.e. number of robots that are no longer available to execute tasks) and with/without assistance mechanisms (i.e. mechanism used to take failed robots outside the transport field).

Table 1: Experimental Designs

Nº	Number of robots	% light-low volume	# faulty robots	With(out) assistance
1	150	90	1 → 5	No
2	150	90	1 → 5	Yes

By altering these input parameters, we have built two experimental designs (see Table 1). Each experimental design contains 120 combinations of values.

The number of robots is fixed to 150, given that, within this configuration, this system poorly tolerates higher number of robots. The percentage of heavy and high volume parcels is kept constant to 10%, which corresponds to the current maximum percentage of heavy and high volume parcels (data provided by PostNL). The number of faulty robots can range from 1 to 5 (from 0.67 to 3.3% failures). The causes of failures are manifold (e.g. software, hardware, energy failures); therefore, although improbable, this number of failures might be observed in practice. Finally, in the last experimental design, we include the assistance mechanism, with robots placed outside the area coming into the field to support the faulty robots.

In these experiments, we consider the impact of these alterations of input parameters on a single key performance indicator, being *throughput* (i.e. number of tasks completed).

It is important to indicate the time of the simulation runs. Assuming that four ticks in our simulation correspond to one second, we will run the simulation for 7200 ticks. However, in order to obtain steady state results, thus eliminating transient state results, we will use a longer run-time (8200 ticks) and delete the results from the first 1000 ticks. By means of observation, we have decided to eliminate the data collected in the first 1000 ticks. Although it is hard to determine a reasonable run length and we are wasting resources / time, the elimination of the initial data provides in all experiments steady results.

This timeframe, assuming the value of 4 ticks per seconds, corresponds to 30 minutes in real-life.

6.1 Results from experiments

By developing disruptive experimental designs, we want to evaluate the robustness of the system.

In the first experimental design, we have limited the number of damaged robots from 1 to 5. This means that during a predefined time interval that we set initially, up to 5 robots stop functioning and remain into the transport field until the end of the shift, interfering with the motion of other robots. Considering that we want to evaluate the effect of disruption on system effectiveness, the temporal interval in which robots fail must be determined ex-ante.

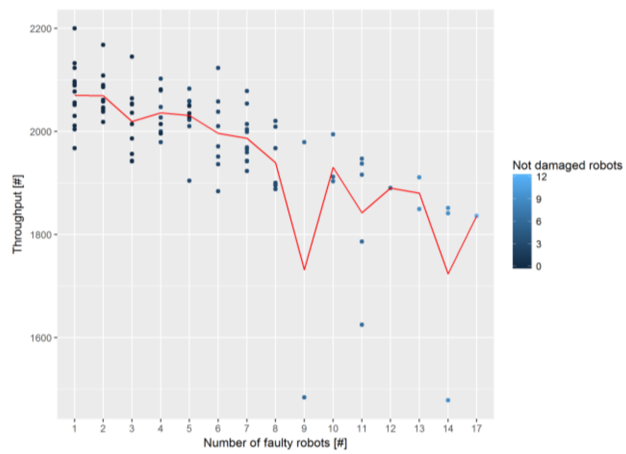

Figure 6: Results from first experimental design

Under these circumstances, the impact of cooperative transport is easy to predict. In fact, when a robot fails in a formation, all the other robots in the formation, although not damaged, are unable to move.

This results in more failures than anticipated. For instance, when we have one single failure, the number of idle robots can be up to four if the failure occurs in a formation; with 2 failures, we can have 2, 5 or 8 idle robots; with 3 failures, we can then have 3, 6, 9 or 12 idle robots, and so forth. In rare cases, robots can fail within the same formation.

Results from this experimental design are shown in the scatterplot in Figure 6. In this plot, on the x-axis we have the number of faulty robots, which includes damaged robots and not damaged robots (i.e. robots that are stuck because one other robot has failed in a formation), and on the y-axis we have the throughput.

As can be observed, the maximum number of faulty robots are 17, which means 12 failures more than the maximum number of failures set initially.

In the plot, we have marked not damaged robots with a different gradient of colour, to highlight the impact of cooperative transport. From previous experiments, we have inferred that in normal conditions, i.e. without failures of robots, the mean throughput is 2098.2 parcels after 7200 ticks.

Comparing the throughput in standard conditions with the throughput in the scatterplot, we can see how the red line, which indicates the mean, gradually decreases with the number of failures.

With one and two failures, the mean throughput is 1.36% less (28 parcels less) in relation to the mean throughput in normal conditions. When the number of failures increases, with up to 4 failures, the mean throughput decreases by around 3% (around 60 parcels less). After 5 failures, the mean throughput drops below 2000 parcels in half an hour.

The effect of cooperative transport on system effectiveness is explicit in this plot, with the mean of throughput dropping vigorously in just half hour. From this plot, it can be also observed that the probability that robots fail in formations is elevated, already with 10% heavy and high volume parcels. Indeed, we have found more failures than initially set in 40 out of 100 measurements (with 20 measurements per scenario). Therefore, we can conclude that cooperative transport has a strong negative impact on system fault tolerance.

In the second experimental design, an assistance mechanism is implemented to address the impact of cooperative transport on fault tolerance, observed in Figure 6. This assistance mechanism consists of other robots placed outside the transport field, which intervene every time a robot fails. The assistance mechanism involves few elementary processes:

- When a robot fails, it communicates with one assisting robot. Same as robot-parcel assignment, also in this case the robot-to-assistant is a 1-to-1 assignment, meaning that one assisting robot can be assigned to only one failed robot and vice versa.
- Once the message is arrived, the assisting robot moves into the transport field to help the failed robots. The assisting robot is not in charge of fixing the failed robots, but they only have to ensure that parcels on failed robots are delivered to the appropriate containers and that these robots are taken out from the field in order to not interfere with the motion of other robots.
- Therefore, once reached the position of the failed robots, the assisting robots check whether these robots have parcels with them or not. If they have a parcel, the assisting robots pick up the failed robots together with their parcels and transport them to the destination of the parcels.
- When the destination of a parcel is reached, this parcel is placed onto the right container. At this point, the assisting robots transport the failed robots outside the transport field for maintenance. However, we assume that within the simulation time, the failed robots can no longer enter the transport field during the considered shift time.

It is important to notice that the scope of the assisting robots is (1) to deliver the parcels to appropriate destinations, (2) to eliminate interferences into the field between failed and not failed robots and (3) to eliminate the impact of cooperative transport on system fault tolerance. As a matter of fact, with regard to the formations of robots, the assisting robots only remove the damaged robots and transport them together with the parcels first to destinations and then outside the field. The other robots that were unable to move, as a consequence of the failure of a robot in formaiton, after the assistance, can again carry out their sorting operations.

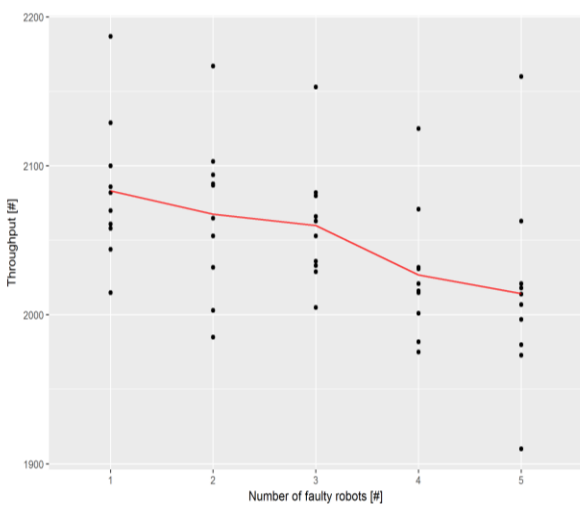

Figure 7: Results from second experimental design

The outcome of the assistance mechanism is explicit when looking at the scatterplot in Figure 7 and compare it with the results obtained in Figure 6. In Figure 7, we can notice that the throughput declines, as a result of the increasing number of damaged robots. However, thanks to the assistance mechanism the impact of cooperative transport vanishes and we have no longer higher number of faulty robots than what initially defined.

Therefore, in this plot the maximum number of faulty robots is exactly 5, while without assistance mechanism this number could arrive up to 20, as earlier explained.

Interestingly, the mean of throughput (red line) is approximately the same as the results obtained in the first experimental design, when the number of faulty robots is below 5. We can conclude that the robustness of the system decreases as a result of the overreliance on individual agents. However, this problem can be addressed by introducing an assistance mechanism.

7. Conclusions

In this scientific article, a solution to the concurrent performance of ST-SR-IA and ST-MR-IA tasks within the same domain of application is proposed.

In the literature paragraph, we have observed that there exists many solutions to address ST-SR-IA tasks, few studies address ST-MR-IA tasks, and hardly any address the combination of ST-SR-IA and ST-MR-IA tasks.

To address this problem, we have used the method suggested by Guerrero and Oliver to address ST-MR-IA tasks, adopting a combination of leader-follower and auction-like algorithm. Adding to this work, we have implemented a solution for the dynamic switch of robot behaviours to address concurrently ST-SR-IA and ST-MR-IA tasks, within the same application. By using this algorithm, when a robot discovers an ST-SR-IA task, it decides to act in a selfish (non-cooperative) manner, transporting the parcels individually to destination. While, when a robot discovers an ST-MR-IA task, it becomes a leader and starts recruiting followers to operate cooperative transport of parcels. This solution is simple, efficient (low idleness) and involves low communication and computation complexity.

However, the overreliance on leaders for the assignment of ST-MR-IA tasks and for the cooperative motion can lead to disruptive situations, as shown in the results of the first experimental design.

To increase system robustness, an assistance mechanism is proposed, which aims at removing robots from the transport field, disengaging trapped (not damaged) robots from the formations. By doing so, the negative effect of cooperative transport on system fault tolerance is eliminated.

8. Further Research

For future work, researchers should develop a decentralized algorithm to address ST-MR-IA tasks using homogenous robots and without sequential assignment of tasks. This algorithm can be then integrated with the suggested reactive behaviour switch method. This can lead to further increasing the robustness of the system.

In this domain of application, an assistance mechanism is required also when using a fully decentralized algorithm for task allocation. Therefore, researchers should focus on developing an assistance mechanism using homogenous robots. In this case, the task of removing robots from the field would again be a combination of ST-SR-IA (e.g. removal and transport of parcels) and ST-MR-IA (e.g. removal and transport of robots) tasks.

References

Consolini, L., Morbidi, F., Prattichizzo, D., & Tosques, M. (2008). Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, *44*(5), 1343-1349.

Desai, J. P., Ostrowski, J. P., & Kumar, V. (2001). Modeling and control of formations of nonholonomic mobile robots. *IEEE transactions on Robotics and Automation*, *17*(6), 905-908.

Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, *23*(9), 939-954.

Guerrero, J., & Oliver, G. (2012). Multi-robot coalition formation in real-time scenarios. *Robotics and Autonomous Systems*, *60*(10), 1295-1307.

Khamis, A., Hussein, A., & Elmogy, A. (2015). Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015* (pp. 31-51). Springer International Publishing.

Lattarulo, V., & Parks, G. T. (2012, June). A preliminary study of a new multi-objective optimization algorithm. In *Evolutionary Computation (CEC), 2012 IEEE Congress on* (pp. 1-8). IEEE.

Mas, I., & Kitts, C. (2010, July). Centralized and decentralized multi-robot control methods using the cluster space control framework. In Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on (pp. 115-122). IEEE.

Parker, L. E., & Tang, F. (2006). Building multirobot coalitions through automated task solution synthesis. Proceedings of the IEEE, 94(7), 1289-1305.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45-77.

Sahota, M. K. (1994, October). Reactive deliberation: An architecture for real-time intelligent control in dynamic environments. In *AAAI* (pp. 1303-1308).

Shehory, O., & Kraus, S. (1995, August). Task allocation via coalition formation among autonomous agents. In IJCAI (1) (pp. 655-661).

Tang, F., & Parker, L. E. (2007, April). A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation. In Robotics and Automation, 2007 IEEE International Conference on (pp. 3351-3358). IEEE.

Yunardi, R. T. (2015, October). Contour-based object detection in Automatic Sorting System for a parcel boxes. In *Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), 2015 International Conference on* (pp. 38-41). IEEE.