Chapter 1

DECENTRALIZED OPTIMIZATION VIA NASH BARGAINING

Steven L. Waslander, Gokhan Inalhan and Claire J. Tomlin

Abstract We present a new method for solving multi-player coordination problems using decentralized optimization. The algorithm utilizes the Nash Bargaining solution as the preferable outcome for all players among the set of Pareto optimal points, under assumptions of convexity. We demonstrate the concept on a multi-agent kinematic trajectory planning problem with collision avoidance. An analysis and numeric comparison of complexity is performed between centralized and decentralized penalty method based optimization. The analysis and the simulations suggest operation regimes where the decentralized method incurs no increase in complexity and even improvement in computation time proportional to the number of players over the centralized method. Experimental results from the MIT rover testbed are presented as well, showing very good correlation between the planned and executed trajectories.

Keywords: Decentralized Optimization, Nash Bargaining Solution, Multi Agent Control

1. Introduction

Multi-agent systems, such as collections of vehicles, autonomous robots and supply chain networks, can often benefit from coordination between agents in achieving system level goals and satisfying inter-agent constraints. In the case of aircraft traffic flow through a constricted airspace, coordination of aircraft trajectories can improve fuel consumption or reduce flight duration while maintaining a minimum safe distance between vehicles at all times. There are many approaches to multi-agent coordination which broadly fall into three categories. Centralized approaches require information about each agent's goals and constraints to be available to a central planner which makes decisions for all agents. Distributed approaches allow individual agents to make decisions, but require some central coordination of the decision process to maintain a complete mathematical model. Finally, decentralized approaches remove the requirements of central coordination and allow individual agents to determine their own actions based on only locally available information. This paper focuses on a decentralized approach to coordination that can provide methods for systems where central coordination is undesirable due to the structure of the problem (ie. competing businesses in a supply chain network) or due to a large number of agents (ie. automobile collision avoidance).

There are many areas of research that touch on aspects of this problem. Decomposition and distributed optimization dates back to results by Benders [Benders, 1962], and was extended to a general class of convex optimization problems by [Geoffrion, 1972], and to non-convex problems by [Tammer, 1987], [Klatte, 1987]. For multiple decision makers, distributed computation of Pareto-optimal solutions has been studied by [Verkama et al., 1994]. However, the result is limited to quasi-concave cost functions and problems with no constraints. [Heiskanen, 1999] provides a decentralized method for calculating Pareto-optimal solutions in multi-party negotiations, using a structure similar to distributed optimization methods. The notion of decentralization optimization for stochastic discrete-event systems has been studied by [Vazquez-Abad et al., 1998]. In addition, team algorithms [Barán et al., 1996] have been developed to solve nonlinear systems of equations in a parallel distributed fashion. We utilize ideas from multi-objective optimization covered by [Miettinen, 1999], [Hillermeier, 2001]. Refer to [Coello, 1998] for an extensive review on this topic. Additionally, we use concepts of decomposition and overlapping given by [Šiljak, 1978] that aid in analyzing large-scale interconnected systems.

Our recent work, [Inalhan et al., 2002a] and [Inalhan et al., 2002b], formulated the multi-agent coordination problem as a cooperative decentralized optimization, and guaranteed that the solution satisfies necessary conditions for Pareto optimality of a centralized formulation. Furthermore, sufficiency conditions for Pareto optimality are met for convex optimization problems, and hence the algorithm is guaranteed to converge to within ϵ of a Pareto optimal solution. In this paper, we select a mutually agreeable solution to convex decentralized optimization problems by constructing an algorithm to search for a specific Pareto optimal point, the Nash Bargaining Solution, as first proposed by John Nash, [Nash, 1950]

The Nash Bargaining Solution was extended to multi-player games with coalitions by Harsani, [Harsani, 1963], and modified for non-convex problems by Conley and Wilkie, [Conley and Wilkie, 1996]. Objections have been raised to one of the axioms needed to define the Nash Bargaining Solution by Kalai and Smorodinsky, who proposed an alternate solution which focuses on global information, [Kalai and Smorodinsky, 1975]. In the decentralized framework, the Nash Bargaining solution remains of interest, however, due to its differentiability and focus on local information.

To the best of our knowledge, this paper presents the following novel results. With the addition of requirements of convexity and communication between all agents, we modify our previous algorithm for decentralized optimization to seek the Nash Bargaining Solution (NBS). We compare, through analysis and simulation, the computational complexity of centralized and decentralized penalty method optimization for non-convex problems. Finally, we demonstrate real-time operation of the decentralized non-convex optimization algorithm on the MIT rover testbed, courtesy of the Aerospace Controls Laboratory under the supervision of Professor Jonathan How.

2. Problem Formulation

Consider a system of p agents, where each agent $i, \in \mathbb{P} = [1 \dots p]$ has associated with it a vector of optimization variables, $x_i \in \mathbb{R}^{n_i}$ with $x = [x_1, \dots, x_p] \in \mathbb{R}^n$. For each agent, we define an independent cost function, $f_i(x_i)$, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}_+$. The *centralized optimization problem* can be defined as,

Definition 2.1 [Centralized Optimization Problem]

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & [f_1(x_1), \dots, f_p(x_p)] \\ \textit{subject to} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{1}$$

where $g : \mathbb{R}^n \to \mathbb{R}^q$, $h : \mathbb{R}^n \to \mathbb{R}^r$ are lists of inequality and equality constraints which can include both local and global requirements. The notation $g^k(\cdot)$ refers to the k^{th} constraint in $g(\cdot)$. In the example of agents as vehicles, the local cost function can be constructed to penalize, for example, deviations from a desired trajectory or fuel consumption. Local constraints can include vehicle dynamics, minimum and maximum control limits, and obstacle avoidance constraints. Global requirements can account for collision avoidance between vehicles, coordinated search requirements or resource allocation among agents. We assume that f_i, g, h are continuously differentiable functions of continuous variables, and that the complete set of constraints is regular [Bertsekas, 1995].

Optimality for the centralized optimization problem is defined using Pareto optimality.

Definition 2.2 [Pareto Optimal Solution] *The vector* $x^{*p} \in \mathbb{F} = \{x \in \mathbb{R}^n | g(x) \leq 0, h(x) = 0\}$ *is a Pareto optimal (minimal) solution of the centralized optimization problem if there exists no* $x \in \mathbb{F}$ *and* $j \in \mathbb{P}$ *such that* $f_i(x_i) \leq f_i(x_i^{*p}) \quad \forall i \in \mathbb{P}$ *and* $f_j(x_j) < f_j(x_j^{*p})$.

The multi-agent coordination problem can also be posed in a decentralized manner. Let us first define an agent i's *Neighborhood* as the set $\mathbb{P}_i \subseteq \mathbb{P}$ of agents j for which there exists a constraint that involves both agents i and j. Intuitively, the notion of neighborhood bounds the scope of interest for an agent to those members of the system that may have an impact on its optimization process. We use the notation $\{x_j\}_i = \{x_j \in \mathbb{R}^{n_j} | j \in \mathbb{P}_i\}$ to refer to the set of optimization variables of all agents j in the neighborhood of agent i. The decentralized framework requires that each agent solve a local optimization based exclusively on information concerning other agents in its neighborhood. The *decentralized optimization problem* can be written as,

Definition 2.3 [Decentralized Optimization Problem]

$$\begin{aligned} \min_{x_i \in \mathbb{R}^{n_i}} \quad & f_i(x_i) \\ \textit{subject to} \quad & g_i(x_i | \{x_j\}_i) \leq 0 \\ & h_i(x_i | \{x_j\}_i) = 0 \end{aligned} \tag{2}$$

Here $g_i(x_i|\{x_j\}_i)$, $h_i(x_i|\{x_j\}_i)$ are lists of inequality constraints and equality constraints on x_i, given that the states of all agents j in the neighborhood of i are held constant. g_i and h_i can be further subdivided into local constraints, $(g_{l_i}(x_i), h_{l_i}(x_i))$, involving only local optimization variables, and interconnected or global constraints, $(g_{g_i}(x_i|\{x_j\}_i)$, $h_{g_i}(x_i|\{x_j\}_i))$ involving the optimization variables of at least one other agent in the neighborhood, $\mathbb{P}_i$. We include similar assumptions as in the centralized formulation, namely that f_i, g_i, h_i are continuously differentiable functions of continuous variables, and that the complete set of constraints is regular. Furthermore, we assume that all interconnected constraints enter each associated local optimization identically.

For the decentralized optimization problem of Eq. 2, we define optimality using the Nash equilibrium.

Definition 2.4 [Nash Equilibrium] *Let* $\mathbb{F}_i = \{x_i \in \mathbb{R}^{n_i} | g_i(x_i|\{x_j\}_i) \leq 0, h_i(x_i|\{x_j\}_i) = 0\}$. *Then* $x^{*ne} \in \mathbb{F}$ *is a Nash equilibrium of the decentralized optimization problem if,* $\forall i \in \mathbb{P}$, *given* $\{x_j\}_i$, $f_i(x_i^{*ne}) \leq f_i(x_i)$, $\forall x_i \in \mathbb{F}_i$.

3. Solution Algorithms

Centralized Algorithms. Two well known techniques for solving the centralized formulation are the Lagrange Multiplier and Penalty methods. In both cases, the local cost function is augmented to include costs which penalize the violation of constraints. If a solution can be found, the Lagrange multiplier method guarantees that constraints will be satisfied, but the method requires a new optimization variable for each constraint. In order to solve the vector optimization defined in Eq. 1, we introduce $\omega \in \mathbb{R}^p$ as a weighting vector of agents costs. The Lagrange multiplier method can then be written as follows (see [Bertsekas, 1995] for a more general formulation).

$$\min_{x\in\mathbb{R}^n} \max_{\lambda\in\mathbb{R}^q_+,\mu\in\mathbb{R}^r} [f_1(x_1)\dots f_p(x_p)]\cdot\omega + \lambda^T g(x) + \mu^T h(x) \tag{3}$$

where the Lagrange multiplier vectors for equality constraints are defined as $\mu \in \mathbb{R}^r$. For inequality constraints, the Lagrange multiplier vector is

$$\lambda \in \mathbb{R}^q_+ \quad \text{where} \quad \begin{cases} \lambda_k \geq 0 & \\ \lambda_k = 0 & \text{if } g^k \text{ is inactive} \end{cases} \qquad \forall k \in \{1\dots q\} \tag{4}$$

With a linear combination of local cost functions, it is not necessarily possible to achieve all Pareto optimal solutions, however, this simplification is required in order to pose an optimization that can be solved using standard non-linear programming methods.

Formulation of the centralized optimization problem via penalty methods allows for a separate treatment of constraints that does not increase the dimension of the optimization problem, but requires iteration of the entire optimization process until convergence. The penalty method assigns costs to the violation of constraints by including a penalty function in the minimization. For comparison to the decentralized penalty method, we use the penalty method formulation only for interconnected constraints. Let each agents' locally feasible region,

$$\mathbb{X}_i = \{x_i \in \mathbb{R}^{n_i} | g_{l_i}(x_i) \leq 0, h_{l_i}(x_i) = 0\} \quad \forall i \in \mathbb{P}$$

and let $\mathbb{X} = \{x \in \mathbb{R}^n | x_i \in \mathbb{X}_i, \forall i \in \mathbb{P}\}$. With equality constraints recast as inequality constraints using slack variables [Boyd and Vandenberghe, 2004], let us define a class of inexact differentiable penalty functions, $P : \mathbb{R}^n \to \mathbb{R}_+$, that penalize all interconnected constraints of a system by,

$$P(x) = \sum_{k=1}^{q_g} \max\left(0, g^k(x)\right)^\gamma \tag{5}$$

where q_g now defines the total number of interconnected constraints in the system and $\gamma \in \mathbb{R}$, $\gamma \geq 2$ defines the order of the penalty function. The centralized optimization problem in penalty method form solves multiple iterations of the following optimization as the penalty parameter, $\beta \in \mathbb{R}_+$, tends to 0.

$$\lim_{\beta \to 0} \Big(\ \min_{x \in \mathbb{X}} \quad [f_1(x_1), \ldots, f_p(x_p)] \cdot \omega + \tfrac{1}{\beta} P(x) \ \Big) \tag{6}$$

The centralized method, using inexact penalty functions, is guaranteed to converge to a solution, given a feasible solution exists and assuming the penalty parameter is selected such that it converges to some value. If the parameter converges to zero, the solution found meets necessary conditions for Pareto optimality. Furthermore, since the optimal solution is feasible and results in $P(x) = 0$, each intermediate solution of the optimization is bounded above by the optimal cost, and thus the optimization cannot become ill-conditioned at any stage of the process.

Decentralized Algorithm. The decentralized algorithm first defined in [Inalhan et al., 2002b] ties a localized penalty method formulation to a bargaining process between agents. A distinction is made between local and interconnected constraints, for in the decentralized approach, interconnected constraints require special treatment. Local penalty functions are defined analogously to Eq. 5, $P_i : \mathbb{R}_i^p \to \mathbb{R}_+$,

$$P_i(x_i | \{x_j\}_i) = \sum_{k=1}^{q_{g_i}} \max \Big(0, g_{g_i}^k (x_i | \{x_j\}_i)\Big)^\gamma \tag{7}$$

where, for each agent, i, q_{g_i} now defines the number of interconnected constraints, respectively. In order to convert from a centralized to a decentralized formulation of the penalty method, a $\beta_i \in \mathbb{R}_+$ pre-multiplier is included in the penalty augmented cost function, $F_i : \mathbb{X}_i \to \mathbb{R}_+$, defined as,

$$F_i = \beta_i \left[f_i(x_i) + \frac{1}{\beta_i} \Big(P_i(x_i | \{x_j\}_i) \Big) \right] \tag{8}$$

This modification is required since local cost functions are no longer bounded above by the optimal solution; only the local optimization variables x_i can be modified by a local optimization. To ensure convergence, the decentralized approach reduces the weight on the local cost at each iteration, instead of increasing the weight on the violation of constraints. This approach, as first defined in [Inalhan et al., 2002b], ensures that the augmented cost functions converge as long as the local penalty parameter, β_i, tends toward 0. Then, for each agent i, the local penalty method formulation for decentralized optimization is,

$$\lim_{\beta_i \to 0} \left(\min_{x_i \in \mathbb{X}_i} \quad \beta_i \left[f_i(x_i) + \frac{1}{\beta_i} \Big(P_i(x_i | \{x_j\}_i) \Big) \right] \right) \tag{9}$$

The decentralized algorithm can proceed in a number of fashions. In sequential form, all agents calculate a desired trajectory in the absence of interconnected constraints. Agent 1 receives the desired solutions from all other agents in its neighborhood and then solves a local optimization problem with the other agents' solutions fixed, to form a new solution set for all p vehicles. This set is passed along to agent 2 who also performs the local optimization and passes on the updated solution set to agent 3, etc. This method causes a bias in the solution against lower numbered agents in favor of higher numbered agents.

In "multi-threaded" form, all agents initially optimize based on the complete set of desired solutions, then pass out solution sets to each other and re-optimize for each solution set received. At each step, an agent could receive up to $p-1$ solution sets for agents in its neighborhood and must select a preferred solutions to ensure the number of solution threads does not expand exponentially. Trimming of solutions threads can be done based exclusively on local information or by considering global preferences defined in terms of other agents' local cost information which can be included in each solution set.

As presented in detail in [Inalhan et al., 2002b], the above algorithm has been shown to converge to a decentralized Nash Equilibrium solution, which is also a Nash Equilibrium of the centralized problem and comes within ϵ of a solution that satisfies the necessary conditions for Pareto optimality. The proof of this assertion hinges on the fact that the bargaining parameters $\beta_i \to 0, \quad \forall i$, which ensures that the augmented cost function does not increase at any step in the process and that the violation of constraints decreases at each step.

The bargaining process inherent in the above algorithm can be driven to an equilibrium solution that satisfies necessary conditions for Pareto optimality through the selection of the bargaining parameter, β_i. Unfortunately, the relationship between β_i and any specific solution is unclear, unlike the centralized case, where variation in the weighting vector, ω, results in Pareto optimal solutions that favor the more heavily weighted agent. Furthermore, we seek to ensure that the solution selected by the algorithm is "fair", that each agent receives an equal amount of the excess in the system, or incurs an equal amount of cost. The range of equilibrium solutions includes solutions where one agent ignores interconnected constraints while the other suffers dearly for it, and it is precisely these situations we wish to avoid by searching for the Nash Bargaining Solution.

4. Nash Bargaining Solution

Axiomatic Foundation. Based on 4 axioms first defined by John Nash in 1950 [Nash, 1950], a unique optimal bargaining solution between two agents can be found if the set of feasible solutions is compact and convex. Let us define such a two-agent bargaining problem by $B = (V_1(x), V_2(x), d, S)$, where $x \in \mathbb{F} = [x_1, x_2]$, is as above with $p = 2$, $V_i : \mathbb{R}^{n_i} \to \mathbb{R}$ are the agents' Von Neumann-Morgenstern utility functions [von Neumann and Morgenstern, 1944], $d = (d_1, d_2) \in \mathbb{R}^2$ is the disagreement point which defines the cost incurred by each agent if no agreement is reached and $S \subseteq \mathbb{R}^2$ is the compact, convex set of all feasible utility pairs that improve on d. We define $x_B^* \in \mathbb{F}$ to be the optimal bargaining solution with optimal utility $s_B^* \in S$.

Nash showed that a unique optimal solution exists which maximizes the product of the utility functions of both players if the following four axioms are satisfied. It was Nash who first chose to use the product of utilities to determine the Nash Bargaining Solution, and although there is no clear interpretation of this construct in relation to the bargaining problem, its simplicity has allowed for its wide adoption and varied uses (see [Osborne and Rubinstein, 1994] for an alternative formulation).

Axiom 4.1 ***Axiom of Rationality****: Each agent prefers the locally optimal solution.*

Axiom 4.2 ***Axiom of Symmetry****: If S is symmetric about the line $V_1 = V_2$, then the optimal bargaining utility lies on that line.*

Axiom 4.3 ***Axiom of Linear Invariance****: Neither scaling nor offset of either utility function affects the resulting bargaining solution.*

Axiom 4.4 ***Axiom of Independence of Irrelevant Alternatives****: If we define $\tilde{B} = (V_1(x), V_2(x), d, \tilde{S})$, where $\tilde{S} \subseteq S$ and the optimal utility, $s_B^* \in \tilde{S}$, then $s_{\tilde{B}}^* = s_B^*$. If S is restricted and yet retains s_B^* of the original problem, then the original optimal bargaining solution remains optimal for the restricted problem.*

Proof Outline (After Nash, [Nash, 1950]). To show existence and uniqueness of an optimal bargaining solution, we invoke properties of compactness and uniqueness of S, respectively. To show that the optimal solution maximizes the product of the utilities of both agents, the following elegant set of arguments was developed based on the four axioms. If both agents are rational they will try to maximize their local utility, V_i. If both utility functions are linearly invariant, then both can be scaled and offset such that $d = (0,0)$ and $s_B^* = (1,1)$. Let $B' = (V_1(x), V_2(x), d, S')$, where S' is augmented to include all points such that the sum of the two utilities is less than 2 (ie. let S' be the triangle formed by the points $\{(0,0),(2,0),(0,2)\}$). Since S' is symmetric, by Axiom 4.2, $s_{B'}^*$ must be on the line $V_1 = V_2$, and thus $s_{B'}^* = (1,1)$. By Axiom 4.4, we see that $s_{B'}^* \in S$, and so is also the optimal solution to the original problem. The final step is to see that s_B^* is the point of maximum product of utility improvements $(V_1(x)-d_1)$, $(V_2(x)-d_2)$, and hence that maximizing the product of utility improvements determines the unique optimal bargaining solution.□

A two dimensional representation of elements of the two-player bargaining problem can be seen in Fig. 1.1.

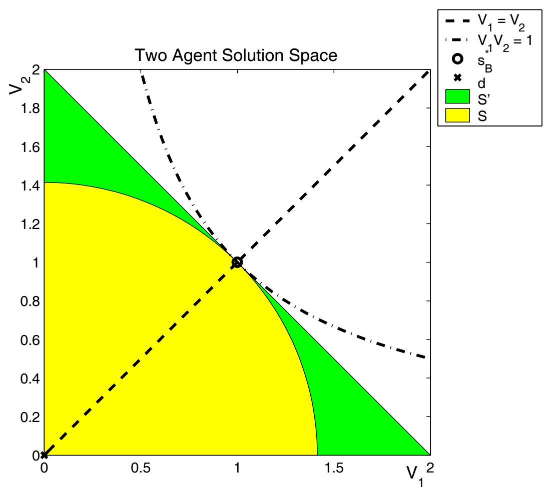

Figure 1.1. Graphical representation of key elements of the Nash Bargaining Solution proof.

Fact 4.1 *The Nash Bargaining Solution is Pareto optimal.*

As defined in Def. 2.2, the Pareto optimal solution requires that no other agent can improve their utility without decreasing the utility of another agent. By Axiom 4.1, both agents must select their locally optimal solution, and by convexity and compactness of the solution space, neither agent can improve their solution from this local optimum without decreasing the other agent's utility.

The same argument can be used for p agents, assuming that the solution space remains convex and compact and the same four axioms hold for all utility functions. The resulting central optimization for determining the p-agent Nash Bargaining Solution (NBS) is,

$$\max_{x\in\mathbb{F}} \left[\prod_{i=1}^{p} (V_i(x) - d_i)\right] \tag{10}$$

Reposing the formulation above as a minimization of cost functions, and adjoining problem constraints using the centralized Lagrangian method of Eq. 3, the NBS is found by minimizing,

$$\min_{x\in\mathbb{R}^n} \max_{\lambda\in\mathbb{R}_+^q,\mu\in\mathbb{R}^r} - \left[\prod_{i=1}^{p} (d_i - f_i(x_i))\right] + \lambda^T g(x) + \mu^T h(x) \tag{11}$$

Likewise, in centralized Penalty Method form, Eq. 6 becomes

$$\lim_{\beta\to 0} \left(\min_{x\in\mathbb{X}} - \left[\prod_{i=1}^{p} (d_i - f_i(x_i))\right] + \frac{1}{\beta} P(x)\right) \tag{12}$$

Necessary Conditions - Centralized Methods. We now turn to a comparison of the necessary conditions for optimality, [Bertsekas, 1995] in order to determine a relationship between the decentralized penalty method and the NBS for a two agent problem. Using the centralized Lagrange multiplier formulation of Eq. 3, the resulting necessary conditions for optimality include,

$$\omega_i \frac{\partial f_i(x_i^*)}{\partial x_i} + \lambda^T \frac{\partial g(x^*)}{\partial x_i} + \mu^T \frac{\partial h(x^*)}{\partial x_i} = 0 \quad \forall i \in \mathbb{P} \tag{13}$$

By contrast, for the optimal solution x^* and the corresponding Lagrange multiplier values $\bar{\lambda}$, $\bar{\mu}$ the NBS necessary conditions can be written explicitly for each agent,

$$\left[\prod_{j\neq i} \left(d_j - f_j(x_j^*)\right)\right] \frac{\partial f_i(x_i^*)}{\partial x_i} + \bar{\lambda}^T \frac{\partial g(x^*)}{\partial x_i} + \bar{\mu}^T \frac{\partial h(x^*)}{\partial x_i} = 0 \quad \forall i \in \mathbb{P} \tag{14}$$

Dividing through by $\prod_{i=1}^{p} (d_i - f_i(x_i^*))$ yields,

$$\left[\frac{1}{(d_i - f_i(x_i^*))}\right] \frac{\partial f_i(x_i^*)}{\partial x_i} + \left[\frac{\bar{\lambda}^T}{\prod_{i=1}^{p} (d_i - f_i(x_i^*))}\right] \frac{\partial g(x^*)}{\partial x_i} + \left[\frac{\bar{\mu}^T}{\prod_{i=1}^{p} (d_i - f_i(x_i^*))}\right] \frac{\partial h(x^*)}{\partial x_i} = 0 \quad \forall i \in \mathbb{P} \tag{15}$$

If the weighting parameters ω_i in the centralized Lagrange multiplier formulation are chosen to be $\frac{1}{\left(d_i - f_i(x_i^*)\right)}$, then the resulting Pareto optimal solution meets the necessary conditions for the NBS. Note that if $d_i = f_i(x_i^*)$, the problem is ill-posed as the optimal solution is disagreement.

Necessary Conditions - Decentralized Methods. From the decentralized formulation of Eq. 9, the resultant necessary conditions become,

$$\beta_i \frac{\partial f_i(x_i^*)}{\partial x_i} + \frac{\partial P_i(x_i^* | \{x_j^*\}_i)}{\partial x_i} = 0 \quad \forall i \in \mathbb{P} \tag{16}$$

The NBS necessary conditions for the Penalty method formulation can be written for each agent as,

$$\left[\prod_{j \neq i} \left(d_j - f_j(x_j^*)\right)\right] \frac{\partial f_i(x_i^*)}{\partial x_i} + \frac{1}{\beta} \frac{\partial P(x_i^* | \{x_j^*\}_i)}{\partial x_i} = 0 \quad \forall i \in \mathbb{P} \tag{17}$$

By Eqs. 5 and 7, the penalty function derivatives, $\frac{\partial P_i(x_i^* | \{x_j^*\}_i)}{\partial x_i}$ and $\frac{\partial P(x_i^* | \{x_j^*\}_i)}{\partial x_i}$ will appear identically in the two sets of necessary conditions, thus for the decentralized algorithm to meet the NBS necessary conditions for optimality, the bargaining parameters, β_i, must be chosen as,

$$\beta_i = \beta \cdot \prod_{j \neq i} \left(d_j - f_j(x_j^*)\right) \quad \forall i \in \mathbb{P} \tag{18}$$

Because the solution space, S, is compact and convex, the decentralized algorithm will converge to within ϵ of a Pareto optimal solution, as both necessary and sufficient conditions for Pareto optimality are satisfied if the solution converges. The optimal cost scaling factor for agent i, $\prod_{j \neq i} d_j - f_j(x_j^*)$, ensures the bargaining process converges to a solution that meets the necessary conditions of the NBS, and since the NBS must be unique, the decentralized algorithm with β_i as defined Eq. 18, will converge to the NBS.

Both centralized and decentralized results provide us with a method for determining the NBS, but are dependent on the optimal costs, and hence must be approximated for implementation. Immediately, the method of successive approximations [Bertsekas, 1993] suggests itself as a means to approximate the desired coefficients. The disagreement point, d_j can be determined by first optimizing locally without interconnected constraints to find the ideal solution for each agent, and then to optimize locally with the ideal solutions for all other agents fixed, which results in a worst case non-cooperative solution for each agent. The NBS can now be found by setting β_i locally, at each iteration, k, of the optimization based on the intermediate optimization results x^{k-1} as follows,

$$\beta_i(k) = \beta(k) \prod_{j \neq i} \left(d_j - f_j(x_j^{k-1}) \right) \quad \forall i \in \mathbb{P} \tag{19}$$

It is important to note the effect of defining bargaining parameters as in Eq. 19 on the communication network between agents. Up to this point, the decentralized framework required that only the current solution be passed by each agent to all others in its neighborhood. In addition, the new bargaining parameter definitions require that each agent receive the current best cost estimate x_j^{k-1} from all other agents in the system, and that each agent execute the update optimization using the same $\beta(k)$. These additional constraints on the communication structure may become restrictive with large numbers of agents, and remain an area for future investigation.

Implementation. The algorithm, as modified by the above discussion, was implemented for the two vehicle collision avoidance problem. Vehicle 1 was located at the point $(6, 0)$ facing west, Vehicle 2 was located at $(0, -7)$ facing north, with desired trajectory defined as straight lines in the forward direction. A quadratic cost was associate with deviation from the desired trajectory, and a collision avoidance constraint required 5 m spacing between the vehicles. A simple kinematic model of an aircraft was used, with control inputs for velocity and turn rate, and a 5-step finite horizon lookahead policy was implemented.

A comparison was made between the original decentralized algorithm, as defined in [Inalhan et al., 2002b], and the same algorithm with bargaining parameters selected as defined in 19. The following graph displays the evolution of the costs for each vehicle for both the original algorithm and the improved NBS inspired algorithm. The NBS method displays much faster convergence to the line between the greedy optimal point (0,0) and the feasible NBS, which will allow for future implementations to use fewer bargaining steps to arrive at the optimal solution. Calculation of the Pareto optimal front and the NBS was performed in a centralized manner using the penalty method for reference.

As mentioned earlier, it is interesting to note that in a single solution thread, the advantage lies in not being the first vehicle in the process. With two vehicles, we can see that if vehicle 1 performs the first optimization given vehicle 2's desired trajectory, then it must select a trajectory that avoids vehicle 2, as required by the bargaining parameter β. Vehicle 2 then performs the next optimization based on vehicle 1's

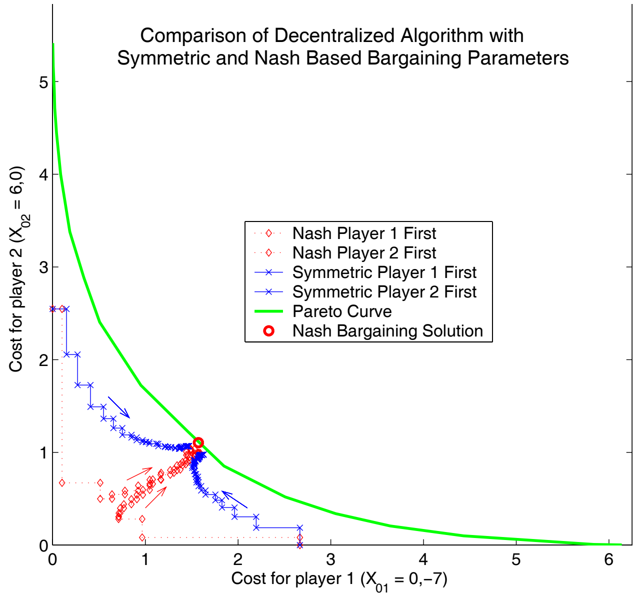

Figure 1.2. Solution space and solution trajectories for NBS-based and symmetric decentralized algorithms. Arrows indicate direction of convergence of algorithm.

solution, and deviates slightly from its desired trajectory, due to a decrease in the value of β which increases the importance of satisfying the interconnected constraints. In Fig. 1.2 and Fig. 1.3 , both threads are displayed for each algorithm and it can be observed that the bargaining process must proceed for some time before this advantage is overcome, unless the Nash-inspired bargaining parameters are used.

5. Complexity Analysis

The complexity of the decentralized algorithm is best compared with an equivalent centralized problem. For this analysis, let p be the number of agents, let A be the number of local control variables, and let B be the number of local constraints.

Nonlinear Program Complexity. The nonlinear optimizations specified above are cast as standard nonlinear programs (NP), where one seeks to find a solution, x, to minimize the global cost function, $F(x)$. Since our problem makes no claim about convexity, we are constricted to finding local minima through an iterative process. The most common algorithm for solving NPs, used in Matlab functions fmincon, fminunc and

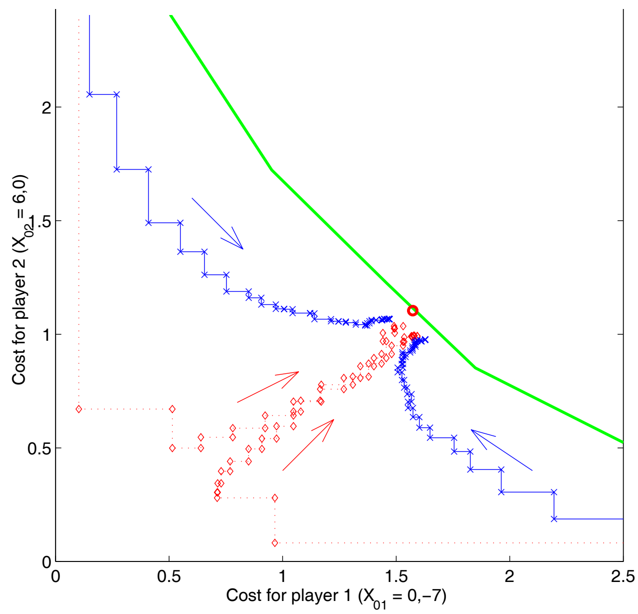

Figure 1.3. Expanded view of the convergence of the algorithms to the Nash Bargaining Solution. Arrows indicate direction of convergence of algorithm.

others for medium scale problems is the sequential quadratic program (SQP), see [Biggs, 1975], [Han, 1977] and [Powell, 1978]. This method iteratively solves a quadratic approximation to the problem based on gradient and Hessian information. The Hessian of the Lagrangian is approximated using the BFGS update, and the quadratic program, (QP), is solved to find a search direction for the original problem. A standard line search is then performed in that direction and the process is repeated. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) update requires the solution of a set of n linear equations ($O(n^3)$), unless sparsity can be exploited. The QP complexity can sometimes be bounded using self-concordant theory, [Nesterov and Nemirovskii, 1989] (when convex and self-concordant cost functions are used), but results in bounds that are orders of magnitude away from average numbers of iterations required. The line search is computationally trivial in comparison to the first two steps. The whole process must also be repeated an uncertain number of times to arrive at the NLP solution, but we assume a fixed problem complexity such that the number of SQP steps is relatively constant with respect to problem size.

Comparison. In order to compare centralized and decentralized methods, first assume that the number of Newton steps required to solve any QP is reasonably constant and equal to K_{qp}, regardless of the order of the problem. Second, assume that the number of iterations needed to solve the NLP using SQP is equal to K_{sqp} and also does not depend on problem size. Furthermore, let us note that the relation between the number of iterations required to converge to a solution using the penalty method and the size of the optimization problem is not well understood, nor is the relation between the number of bargaining steps to converge to a solution in the decentralized problem and the number of vehicles bargaining. We therefore introduce variables K_b for the number of bargaining steps used in the decentralized problem and K_p for the number of penalty iterations in the centralized problem as parameters that can be varied in simulation.

The centralized approach with a fixed number of penalty method iterations results in a computational complexity of,

$$O(K_p \times p^3(A+B)^3 \times K_{qp} \times K_{sqp}) = O(K_p \times p^3(A+B)^3) \qquad (20)$$

Likewise, the decentralized approach solves

$$O((A+B)^3 \times K_{qp} \times K_{sqp}) \qquad (21)$$

at each of p vehicles for each of $p-1$ received solutions, and then repeats this process K_b times. The resulting algorithmic complexity is

$$O(K_b \times p^2(A+B)^3 \times K_{qp} \times K_{sqp}) = O(K_b \times p^2(A+B)^3) \qquad (22)$$

Hence, based on the assumptions made above and ignoring the effect of K_p and K_b on the quality of the solution, the result states that the decentralized approach outperforms the centralized approach as the number of agents grows, which is due to its ability to exploit the inherent problem structure.

Simulation Results. In a multi-vehicle collision avoidance simulation, both algorithms were run with varying values for the number of vehicles, the number of bargaining steps/penalty steps and the number of control inputs and constraints. The simulation calculated finite horizon lookahead control policies for 30 time steps, based on quadratic costs for deviation from the desired straight-line trajectory, and 5 mile collision avoidance constraints for the entire horizon. The resulting simulation times are listed in Tbl. 1.1 below.

For the decentralized algorithm, the computation times grew on the order of 0.7 with respect to K_b, which shows that the optimizations

Decentralized Computation Times

Local Variables	10	14	20	10	14	20	10	14	20
Bargaining Steps	5	5	5	10	10	10	15	15	15
No. Vehicles									
2	65	127	267	106	178	377	136	265	407
3	139	292	745	197	494	1360	294	631	1812
4	318	773	1979	561	1224	2986	796	1586	4218

Centralized Computation Times

Local Variables	10	14	20	10	14	20	10	14	20
Penalty Steps	5	5	5	10	10	10	15	15	15
No. Vehicles									
2	147	315	719	220	458	1048	270	591	1296
3	466	1054	2948	682	1581	4414	787	2002	5020
4	980	2300	7090	1443	3554	9225	1716	4369	12637

Table 1.1. Simulation Times (s) - Decentralized and Centralized Methods: 30-period collision avoidance problem

proceeded more quickly as K_b grew, which is most likely due to the fact that the number of steps required for convergence of the SQP algorithm reduced as the bargaining parameter, β converges to zero.

The centralized simulation results concur with the predicted complexity analysis, with the exception of the number of Penalty method iterations. Computation time varied as Q^3 and $\sqrt{K_p}$. The acceleration in the computation time for high number of iterates is due to the simplification of the problem as the iterations proceed, but at a faster rate than for the decentralized case. If the change in β is small, the optimization is nearly identical to the previous step, and so, with the solution of the previous iteration as initial estimate, almost no optimization is necessary.

The improvement in computation time of the decentralized algorithm over the centralized method was further investigated with a simplified problem of only one time step, such that initial conditions for each optimization were identical for both methods. The problems were posed such that significant optimization was necessary (the interconnected constraints were active in the optimal solution), and systems of 3-6 vehicles were simulated to get a better picture of the relation between the number of vehicles and computation time. The results, as displayed in Fig. 1.4

showed p growth for the decentralized case, as predicted from the analysis, and p^3 for the centralized problem.

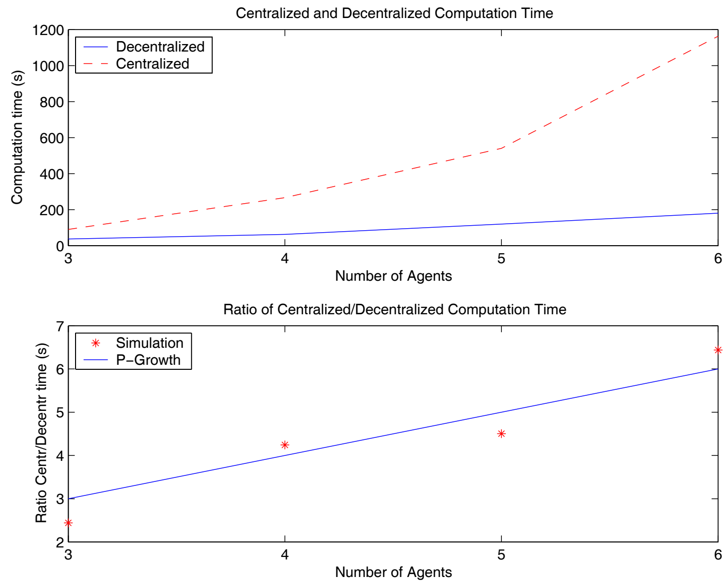

Figure 1.4. Simulation time comparison of centralized and decentralized algorithms for 3-6 vehicles, 5 bargaining iterations and 10 step finite horizon lookahead control.

We should note at this point that nonlinear optimization tools such as Stanford's SNOPT [Gill et al., 2002] can detect and exploit sparsity in any given optimization problem, and may be able to recover most or all of the gains in computation presented here. The decentralized algorithm is inherently designed around the problem structure, however, and so should maintain the advantage.

6. Testbed Validation

Working with the MIT Rover Testbed courtesy of Professor Jonathan How and the MIT Aerospace Controls Laboratory [Richards et al., 2003], we implemented a three-vehicle collision avoidance scenario. The rovers are equipped with an indoor positioning system with cm-level accuracy and on board Sony Vaio laptops which communicate with a ground station via wireless ethernet, see Fig. 1.5. The decentralized algorithm was implemented using 5 step, discretized, receding horizon control with 1 meter collision avoidance constraints between vehicles. The local optimizations were performed using Matlab's fmincon nonlinear optimization program, and new way points were passed to the vehicles at 2 second intervals. The results displayed in Fig. 1.7 show the promise of implementing the proposed decentralized algorithm in real time on real hardware, and validates future extensions of the algorithm to multiple vehicle testbeds and real world applications.

Figure 1.5. MIT Rover Testbed closeup with on board laptop and position sensor visible, courtesy of Jonathan How

Figure 1.6. MIT Rover Testbed in action performing 3 vehicle collision avoidance

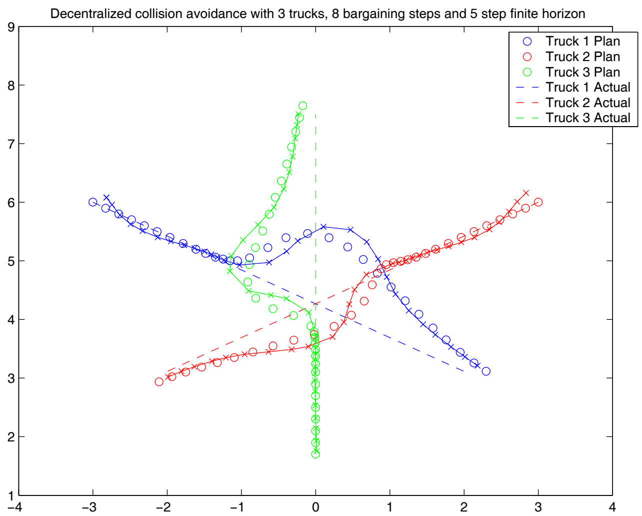

Figure 1.7. MIT Rover Testbed Results, 3 vehicle traffic circle solution

References

[Barán et al., 1996] Barán, B., Kaszkurewicz, E., and Bhaya, A. (1996). Parallel asynchronous team algorithms: Convergence and performance analysis. *IEEE Transactions on Parallel and Distributed Systems*, 7(7):677–688.

[Benders, 1962] Benders, J. F. (1962). Partioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*.

[Bertsekas, 1993] Bertsekas, D. P. (1993). *Dynamic Programming*, volume 1. Athena Scientific, Belmont, Mass., 2nd edition.

[Bertsekas, 1995] Bertsekas, D. P. (1995). *Nonlinear Programming*. Athena Scientific, Belmont, Mass., 2nd edition.

[Biggs, 1975] Biggs, M. (1975). *Towards Global Optimization*, chapter Constrained Minimization Using Recursive Quadratic Programming. North-Holland.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, England.

[Coello, 1998] Coello, C. A. C. (1998). An updated survey of GA-based multiobjective optimization techniques. Technical Repot RD-98-08, Laboratorio Nacional de Informtica Avanzada (LANIA), Xalapa, Veracruz, Mexico.

[Conley and Wilkie, 1996] Conley, J. P. and Wilkie, S. (1996). An extension of the Nash bargaining solution to non convex problems. *Games and Economic Behavior*, 13(1):26–38.

[Geoffrion, 1972] Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4).

[Gill et al., 2002] Gill, P. E., Murray, W., and Saunders, M. A. (2002). *User's Guide for SNOPT Version 6', A Fortran Package for Large Scale Non Linear Programming*.

[Han, 1977] Han, S. (1977). A globally convergent method for nonlinear programming. *J. Optimization Theory and Applications*, 22:297.

[Harsani, 1963] Harsani, J. C. (1963). A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–200.

[Heiskanen, 1999] Heiskanen, P. (1999). Decentralized method for computing Pareto solutions in multi-party negotiations. *European Journal of Operational Research*, 117(3):578–590.

[Hillermeier, 2001] Hillermeier, C. (2001). *Nonlinear Multiobjective Optimization: A generalized homotopy approach*. Birkhauser Verlag, Basel.

[Inalhan et al., 2002a] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J. (2002a). Decentralized optimization, with application to multiple aircraft coordination. SUDAAR 759, Stanford, Palo Alto, CA.

[Inalhan et al., 2002b] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J. (2002b). Decentralized optimization, with application to multiple aircraft coordination. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas.

[Kalai and Smorodinsky, 1975] Kalai, E. and Smorodinsky, M. (1975). Other solutions to Nash's bargaining problem. *Econometrica*, 43(3):513–518.

[Klatte, 1987] Klatte, D. (1987). Strong stability of stationary solutions and iterated local minimizations. In et. al., J. G., editor, *Parametric Optimization and Related Topics*, volume 35 of *Mathematical Research*, pages 119–136. Akademie-Verlag.

[Miettinen, 1999] Miettinen, K. M. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic.

[Nash, 1950] Nash, J. F. (1950). The bargaining problem. *Econometrica*, volume 18(issue 2):pg 155–162.

[Nesterov and Nemirovskii, 1989] Nesterov, Y. and Nemirovskii, A. (1989). Self-concordant functions and polynomial time methods in convex programming. Moscow. USSR Academy of Science, Central Economic and Mathematical Institute.

[Osborne and Rubinstein, 1994] Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory.* MIT Press, Cambridge, Massachusetts.

[Powell, 1978] Powell, M. (1978). Fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis*, 630. Lecture Notes in Mathematics.

[Richards et al., 2003] Richards, A., Kuwata, Y., and How, J. (2003). Experimental demonstrations of real-time MILP control. In *Proceeding of the AIAA Guidance, Navigation, and Control Conference*.

[Tammer, 1987] Tammer, K. (1987). The application of parametric optimization and imbedding to the foundation and realization of a generalized primal decomposition approach. In et. al., J. G., editor, *Parametric Optimization and Related Topics*, volume 35 of *Mathematical Research*, pages 376–386. Akademie-Verlag.

[Vazquez-Abad et al., 1998] Vazquez-Abad, F. J., Cassandras, C. G., and Julka, V. (1998). Centralized and decentralized asynchronous optimization of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 43(5):631–655.

[Verkama et al., 1994] Verkama, M., Ehtamo, E., and Hamalainen, R. P. (1994). On distributed computation of Pareto solutions in n-player games. Research Report A53, Helsinki University of Technology, Systems Analysis Laboratory.

[von Neumann and Morgenstern, 1944] von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior.* John Wiley and Sons, New York.

[Šiljak, 1978] Šiljak, D. D. (1978). *Large-Scale Dynamic Systems: Stability and Structure.* North-Holland, New York,NY.