15-780: Graduate AI
Computational Game Theory

Michael Benisch

Carnegie Mellon University,
School of Computer Science

mbenisch@cs.cmu.edu

Outline

- What is a game?
- Multi-Objective Optimization vs. Game Theory
- Importance of Game Theory in AI
  - Helps agents select strategies
  - Guarantees about artificially designed mechanisms
  - Automated analysis of strategic models
  - Games in the real world
- Solving games with AI
  - Computing Nash equilibria
  - Complexity results on solving games
  - Alternative solution concepts

mbenisch@cs.cmu.edu

Outline (cont'd)

- Building games with AI
  - Mechanism design problem and Revelation Principle
  - Game theoretic properties of auctions: 1^{st} price, 2^{nd} price, eBay
  - Implementation in dominant strategies
  - Vickrey-Clarke-Groves Mechanism
  - Automated Mechanism Design

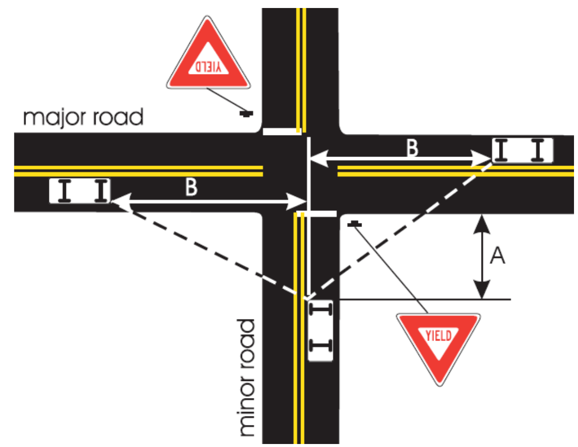
mbenisch@cs.cmu.edu

Background

mbenisch@cs.cmu.edu

What is a Game?

- A game is a multi-agent model of the relationships between agent's actions and incentives.
  - When agents are self-interested the game models an optimization process
  - Games can have underlying probabilistic models to describe uncertain outcomes

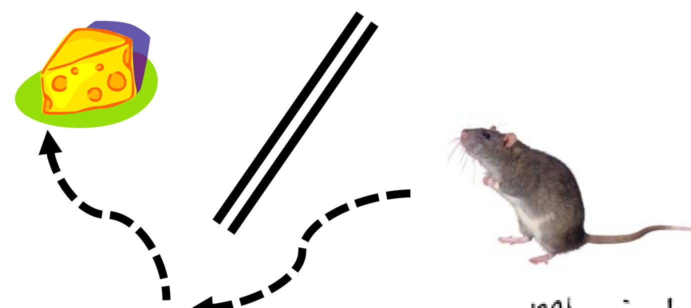


Questions asked about a game

- How should an agent behave?

- What is the most likely state the game will settle in?

- Can the game be designed to incentivize specific actions?



mbenisch@cs.cmu.edu

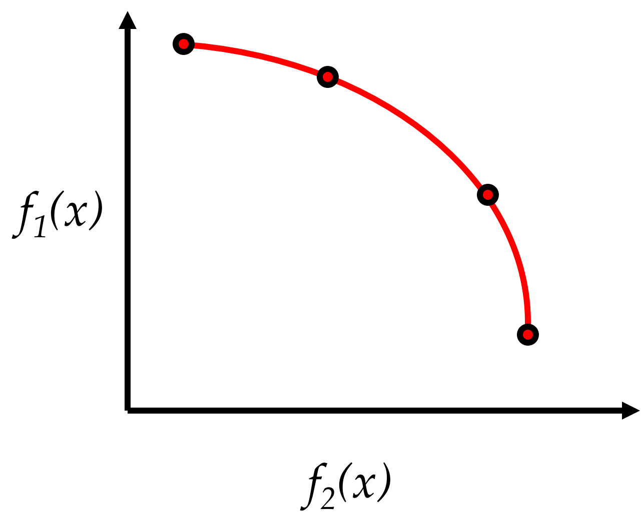
Multi-Objective Optimization

- Class of optimization models involving simultaneously optimizing multiple objectives:

$$\max_{x} \begin{Bmatrix} f_1(x) \\ f_2(x) \\ \ldots \\ f_n(x) \end{Bmatrix}$$

Optimization Model

Solution: Pareto-optimal curve (set of points where each obj. fn. cannot grow larger without decreasing another).



Multi-Objective Optimization vs. Game Theory

- Games are *similar* to multi-objective optimization models, differences are:
  - Each objective function is *owned* by a different agent.
  - The decision variables are partitioned into those controlled by the owner of each objective function.

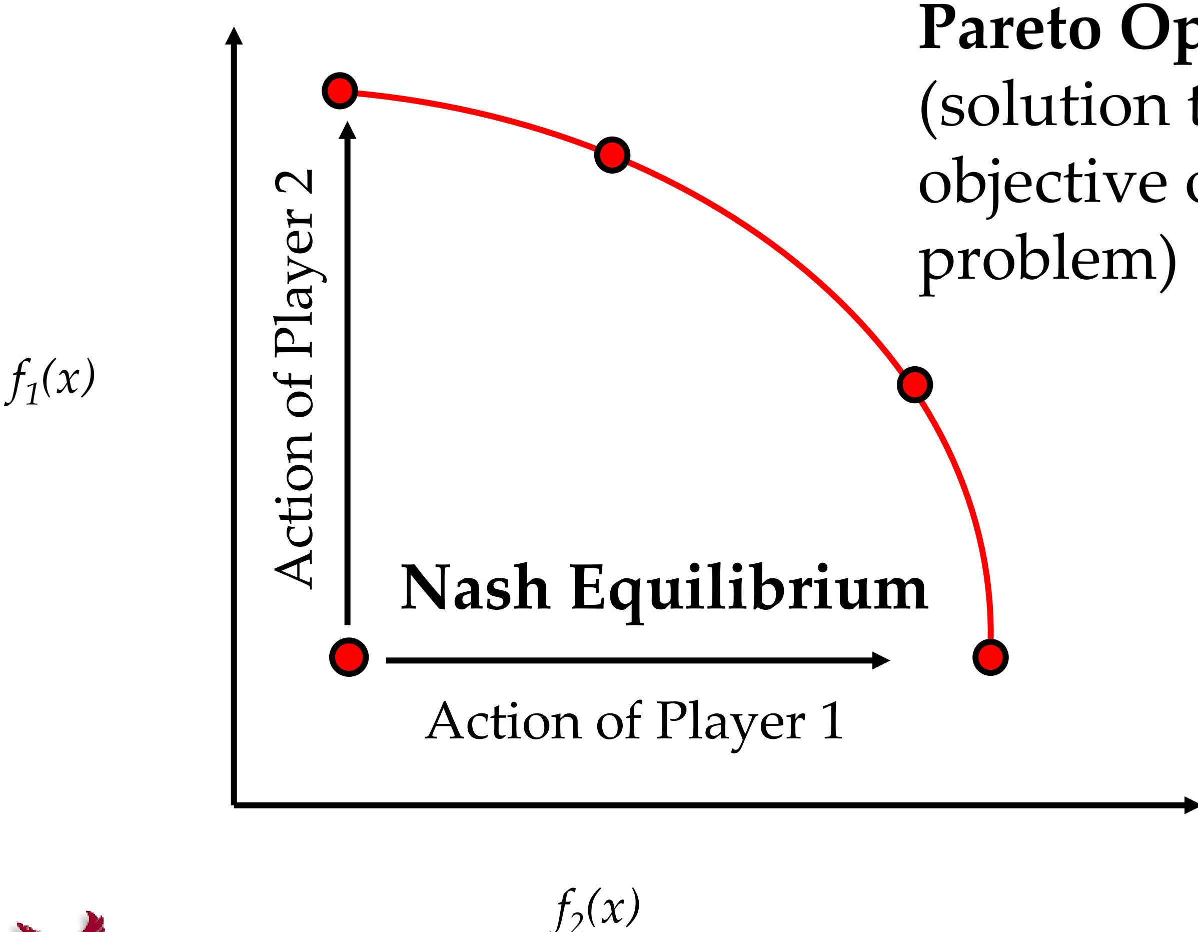
$$f_1(x)$$
$$f_2(x)$$
$$\ldots$$
$$f_n(x)$$

- Agent 1 owns f_1 and controls variables $x_1, \ldots, x_k$

$$x = \{x_1, x_2, \ldots, x_k, \ldots, x_m\}$$

mbenisch@cs.cmu.edu

Multi-Objective Optimization vs. Game Theory (cont'd)



Pareto Optimal Curve (solution to multi-objective optimization problem)

mbenisch@cs.cmu.edu

AI + Game Theory

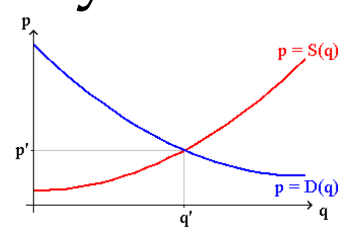
- Help agents select strategies

- Help design games that have certain properties

Name Your Own Price SM Per Round-trip Ticket: $ .00

- Help analysts understand a system

Gambit



CMU School of Computer Science

mbenisch@cs.cmu.edu

Real World Games

- **Games related to warfare**
  - Pursuit and evasion: dogfights, missiles, troops
  - Strategic resource deployment: troops, weapons
- **Games related to economics**
  - Auctions: FCC Spectrum, Google keywords
  - Buying/Selling: resource procurement, stock market, dynamic pricing
- **Games related to networks**
  - Network formation: social, corporate, P2P
  - Graphical games: dependency of player actions is described by network between players
- **Recreational games**
  - Perfect information: chess, checkers, go
  - Limited information: poker, football, video games

mbenisch@cs.cmu.edu

Solving Games with AI

Review: Notation

- **Agent = player:** set of all players is $N = \{1,\ldots,n\}$
- **Action = move:** choice that an agent can make at a point in the game
- **Strategy a_i:** mapping from distinguishable states of the game to actions
- **Strategy set S_i:** strategies available to agent i

mbenisch@cs.cmu.edu

Review: Notation

- **Strategy profile** $S = \{s_1, \ldots, s_n\}$**:** one strategy per agent
- **Utility function** $u_i(S)$**:** mapping from strategy profiles to utilities for player i
- **Opposing profile,** S_{-i} : strategies of agents other than i (in general the notation $-i$ excludes i)

mbenisch@cs.cmu.edu

Review: Key Concepts

- **Normal Form (Matrix, Simultaneous) Game**:
  - Outcome functions are matrices for each player
  - A player's matrix indicates his utility for playing each possible action against any opponent profile.

- **Example NFG**: Prisoner's Dilemma

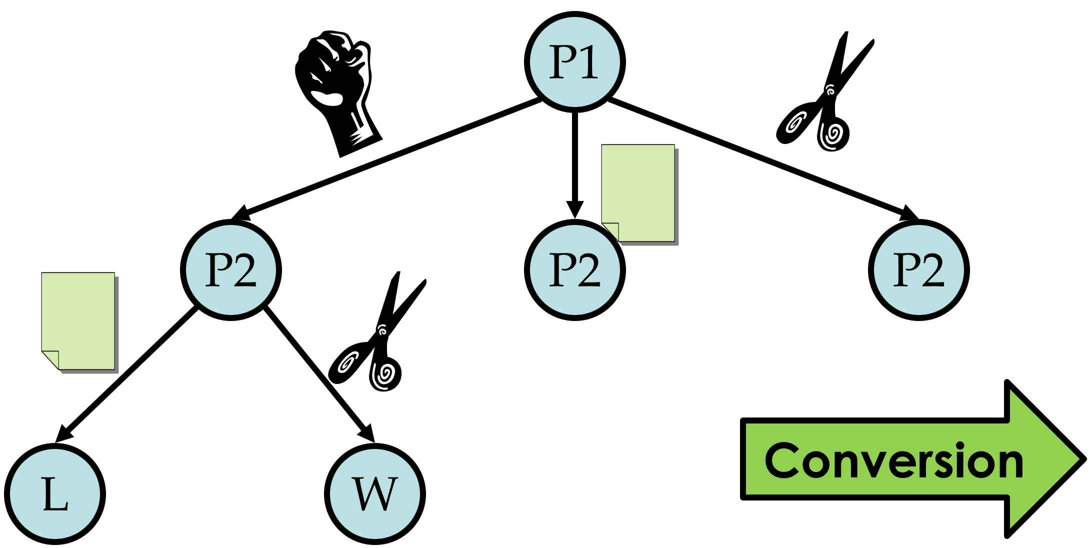
	C	D
c	(3,3)	(0,5)
d	(5,0)	(1,1)

mbenisch@cs.cmu.edu

Review: Key Concepts

- **Extensive Form Game**: provides additional tree structure to game, allowing for players to take turns sequentially (also called sequential form).

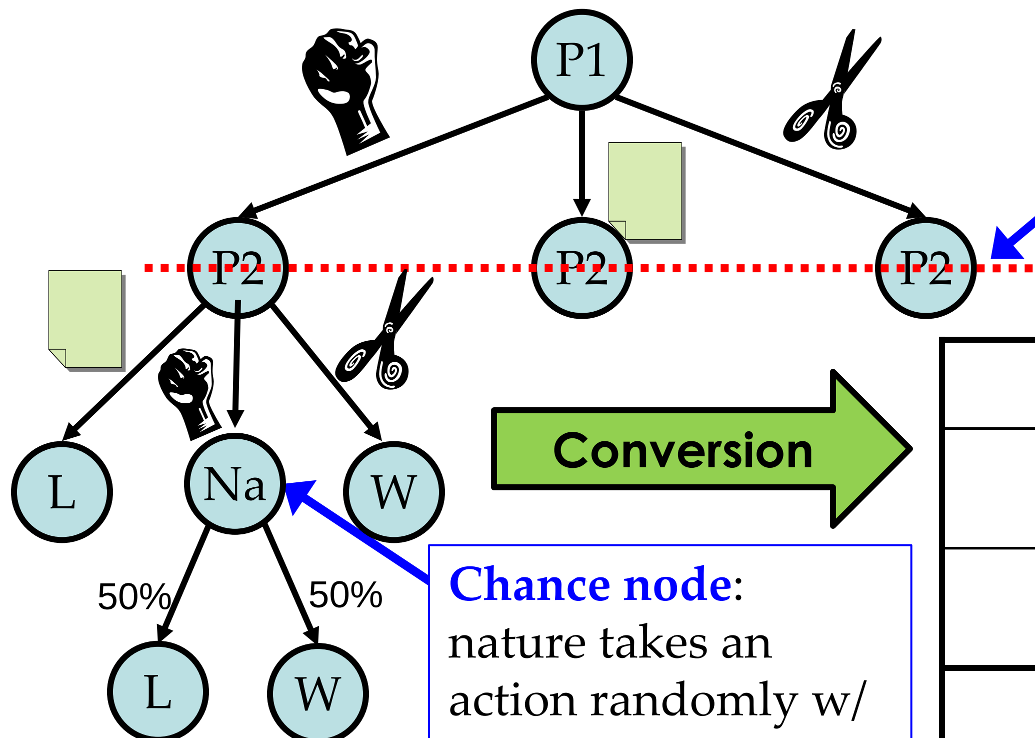
- **Example EFG:** Iterative Rock-Paper-Scissors



	RRR	…	SSS
r	(0,0)	…	(1,-1)
p	(1,-1)	…	(-1,1)
s	(-1, 1)	…	(0,0)

Review: Key Concepts

- **EFG Imperfect information and Chance nodes**: players cannot observe all prior moves and some moves are made by "nature"



Information set: player 2 does not know which node he is in.

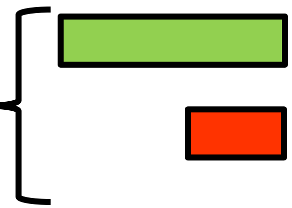
Chance node: nature takes an action randomly w/ specified probability

	R	P	S
r	(1,-1) 50% (-1,1) 50%	(-1,1)	(1,-1)
p	(1,-1)	(1,-1) 50% (-1,1) 50%	(-1,1)
s	(-1, 1)	(1,-1)	(1,-1) 50% (-1,1) 50%

Review: Key Concepts

- **Mixed strategy (profile):** a randomized strategy that specifies probabilities with which to take each action.

[0.75, 0.25]



	C	D
c	(3,3)	(0,5)
d	(5,0)	(1,1)

- **Best response**: the action corresponding to the highest (expected) utility given the actions of other players.

  - **Proposition:** any player has a pure strategy best response to every opponent (mixed) profile

	C	D
c	(3,3) →	(0,5)
d	(5,0) →	(1,1)

mbenisch@cs.cmu.edu

Solving Games

- **Solving a game**: predicting (or suggesting) agent behavior and the resulting outcome(s) of the game.

- **Solution concept:** the principle by which agents are assumed to act.

  - Default concept is **Nash equilibrium**: players will settle into a profile when they cannot unilaterally improve.

	C	D
c	(3,3)	(0,5)
d	(5,0)	(1,1)

mbenisch@cs.cmu.edu

Finding Nash Equilibria

- **Existence:** Nash proved at least one equilibrium in (potentially) mixed strategies always exists
  - **Proof sketch:** Uses Brouwer's fixed point theorem which states that every "regular" *n*-D function has at least one fixed point x such that $f(x) = x$.
- **Zero-sum games:** linear programming solution
- **Pure-strategy equilibrium:** one strategy per player
  - **Perfect Info Extensive-form games:** mini-max search
  - **Normal-form games:** enumeration of all combinations

mbenisch@cs.cmu.edu

Algorithms for Finding Nash Equilibria

- **Supports:** a set of strategies with non-zero probability in some mixture
  - **Proposition**: Knowing supports of an NE allows computation of strategies in polynomial time by solving a feasibility problem (which is linear for 2-players).
  - **Constraint equations:** find mixtures p_i over supports in S_i such that all players are indifferent between the strategies in their supports

mbenisch@cs.cmu.edu

Algorithms for Finding Nash Equilibria

- **Feasibility problem:** find p and v such that,

$$\forall i \in N, a_i \in S_i : \quad \sum_{a_{-i} \in S_{-i}} p(a_{-i}) u_i(a_i, a_{-i}) = v_i$$

Agents are indifferent between all strategies in supports

$$\forall i \in N, a_i \notin S_i : \quad \sum_{a_{-i} \in S_{-i}} p(a_{-i}) u_i(a_i, a_{-i}) \leq v_i$$

Strategies outside of supports are worse

$$\forall i \in N : \quad \sum_{a_i \in S_i} p_i(a_i) = 1$$

$$\forall i \in N, a_i \in S_i : \quad p_i(a_i) \geq 0$$

$$\forall i \in N, a_i \notin S_i : \quad p_i(a_i) = 0$$

Mixture is valid (sums to 1) and no strategies out of supports are included

mbenisch@cs.cmu.edu

Algorithms for Finding Nash Equilibria

- Lemke-Howson Algorithm [1967]
  - Pivoting based algorithm similar to Simplex; very fast in practice
  - Strategies are pivoted into and out of supports

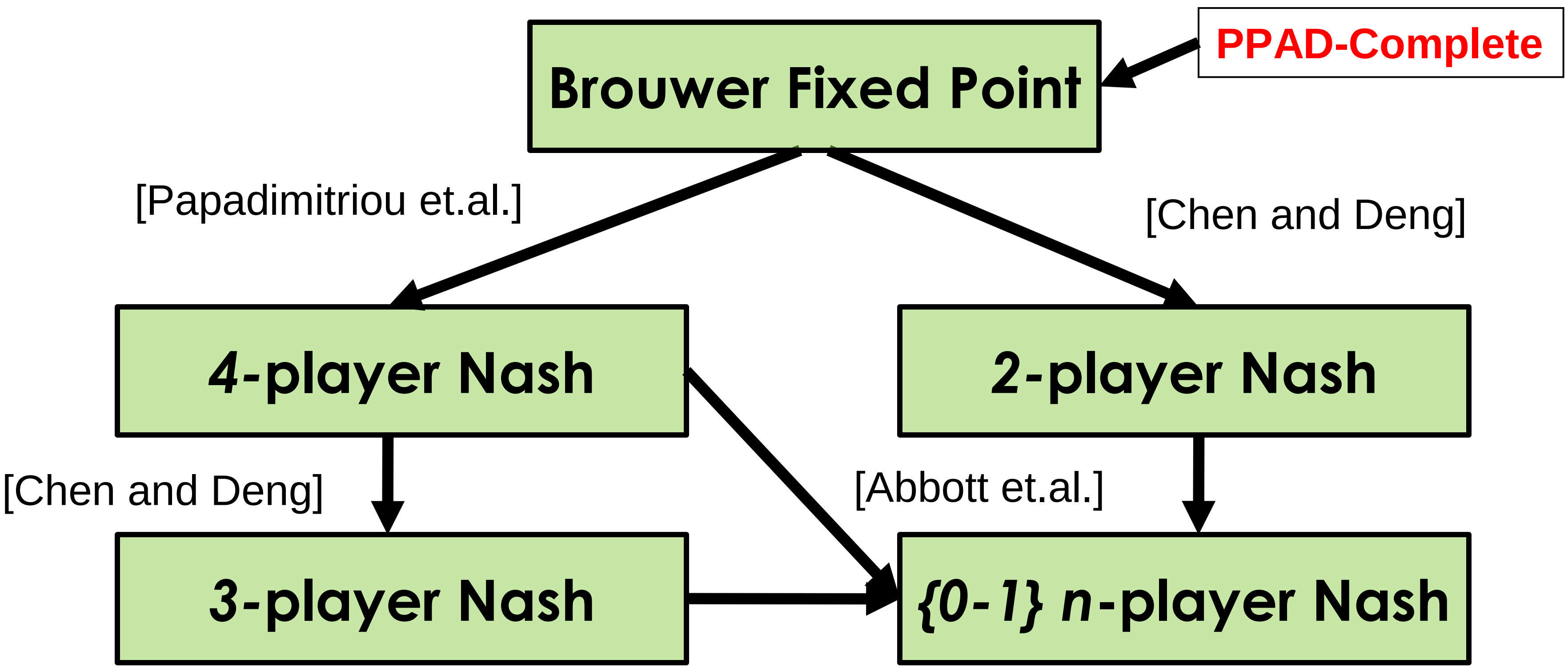
- Porter, Nudelman, and Shoam (PNS) [AAAI-04]
  - Treat support for each player as $\{0,1\}^{|S|}$ vector
  - Brute-force support enumeration algorithm
  - Can be generalized beyond 2-players (with nonlinear program)

- Gilpin, Conitzer, and Sandholm: MIP Nash [AAAI-05]
  - Mixed-Integer Programming (MIP) formulation
  - **Main insight:** regret is 0 in equilibrium

- **All worst-case exponential time in size of game.**

Finding Nash Equilibria: Complexity

- General sum normal form reductions (last 2 years)



Finding Nash Equilibria: Complexity

- **Theorem:** Finding a single NE is PPAD-Complete even in 2-Player games with binary payouts.
  - **PPAD:** Subclass of TFNP, which is a collection of NP-Complete decision problems which are known to be true
  - Other TFNP problems: factoring integers, solvable CSPs

mbenisch@cs.cmu.edu

Finding Nash Equilibria: Complexity

- Deciding whether a "good" equilibrium exists is NP-complete from SAT reduction [Conitzer and Sandholm]:
  - ❑ equilibrium with high social welfare
  - ❑ Pareto-optimal equilibrium
  - ❑ equilibrium with high utility for a given player
  - ❑ equilibrium with high minimal utility

- Also NP-complete (same reduction):
  - ❑ Does more than one equilibrium exists?
  - ❑ Is a given strategy ever played in any equilibrium?
  - ❑ Is there an equilibrium where a given strategy is never played?
  - ❑ Is there an equilibrium with >1 strategies in the players' supports?

mbenisch@cs.cmu.edu

Criticisms of Nash equilibrium

- **Not necessarily unique**: some games have multiple NEs, which will agents settle into?
  - Social-welfare maximizing?
  - Pareto-optimal?

- **Can be hard to compute**

- NE is **not consistent**
  - One player can unilaterally move system from one equilibrium to another

mbenisch@cs.cmu.edu